

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΣΧΕΔΙΟ ΠΑΡΟΥΣΙΑΣΗΣ
ΑΤΟΜΙΚΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Μάιος 2019

Ατομική Διπλωματική Εργασία

**ΑΝΙΧΝΕΥΣΗ ΟΧΗΜΑΤΩΝ ΣΕ ΕΝΑΕΡΙΕΣ ΛΗΨΕΙΣ
ΜΕ ΤΗΝ ΧΡΗΣΗ ΣΥΝΕΛΙΚΤΙΚΩΝ ΔΙΚΤΥΩΝ**

Παρασκευή Χριστοδούλου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2019

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανίχνευση Οχημάτων σε Εναέριες Λήψεις με την Χρήση Συνελικτικών Δικτύων

Παρασκευή Χριστόδουλου

Επιβλέπον Καθηγητής

Κωνσταντίνος Παττίχης

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2019

Ευχαριστίες

Εκφράζω τις θερμές μου ευχαριστίες μου στον επιβλέπον καθηγητή μου Κωνσταντίνο Παττίχη για την βοήθεια, καθοδήγηση και υποστήριξη κατά την εκπόνηση της παρούσας διπλωματικής εργασίας. Επίσης, ευχαριστώ πάρα πολύ τον καθηγητή Ανδρέα Παναγίδη για την καθοδήγηση, υποστήριξη και τις συμβουλές που μου προσέφερε κατά τη διάρκεια της υλοποίησης του συστήματος της εν λόγω διπλωματικής.

Θέλω να ευχαριστήσω τους γονείς μου για την στήριξη που μου παρείχαν κατά την διάρκεια των προπτυχιακών μου σπουδών στο πανεπιστήμιο και κατά την διάρκεια της εκπόνησης της παρούσας διπλωματικής εργασίας.

Ευχαριστώ πολύ τον παππού μου, την γιαγιά μου, τους θείους/θείες και τους υπόλοιπους συγγενείς και φίλους για το ενδιαφέρον τους κατά την διάρκεια των σπουδών μου και κατά την διάρκεια της εκπόνησης της παρούσας διπλωματικής εργασίας.

Ευχαριστώ θερμά τους φίλους μου Έλλη, Έλενα, Γεωργία και Νίκο για την υποστήριξη τους κατά την διάρκεια της εκπόνησης της παρούσας διπλωματικής εργασίας.

Περίληψη

Στην παρούσα διπλωματική εργασία μελετήθηκαν πολλές και διαφορετικές αρχιτεκτονικές συνελκτικών δικτύων και πως μπορούν να ανιχνεύσουν οχήματα σε εναέριες λήψεις και να προβλέψουν την ακριβή τους τοποθεσία. Ένα επίπεδο συνέλιξης σε ένα συνελκτικό δίκτυο έχει την ικανότητα να μαθαίνει συγκεκριμένα χαρακτηριστικά ενός αντικειμένου π.χ τις ακμές. Ακολουθήθηκε μια συγκεκριμένη μεθοδολογία κατά την υλοποίηση της διπλωματικής. Αυτή αποτελείτο από τα εξής βήματα: Επιλογή Βάσης Δεδομένων, Εξαγωγή Χαρακτηριστικών Οχημάτων, Προ επεξεργασία Δεδομένων, Διαχωρισμός Βάσης Δεδομένων σε Δεδομένα Εκπαίδευσης και Επαλήθευσης, Εκπαίδευση Κατηγοριοποιητών Οχημάτων με την χρήση Transfer Learning (προ εκπαιδευμένων βαρών ImageNet και εγχείρηση συνελκτικών δικτύων), της τεχνικής GridSearch μαζί με τον αλγόριθμο εκπαίδευσης Logistic Regression και διαφορετικούς αλγόριθμους ενημέρωσης συναπτικών βαρών με διαφορετικές παραμέτρους και τιμές παραμέτρων. Ακολούθως, τα μοντέλα που παράχθηκαν από τους κατηγοριοποιητές δόθηκαν στους ανιχνευτές οχημάτων οι οποίοι με την χρήση πυραμίδων εικόνων (για ανίχνευση οχημάτων διαφορετικής κλίμακας), μετακινούμενων παραθύρων (για ακριβή εντοπισμό της θέσης των οχημάτων), συγκεκριμένων παραμέτρων εισόδου και του αλγόριθμου Logistic Regression πρόβλεπαν την πιθανότητα των αντικειμένων της εικόνας να είναι οχήματα. Στη συνέχεια, παρουσιάζονται τα οχήματα που προβλέφθηκαν πως υπάρχουν μέσα στην εικόνα και γύρω από κάθε όχημα τοποθετείται ένα κουτί (για εύκολο εντοπισμό του αντικειμένου και της θέσης του). Το αποτέλεσμα της παρούσας διπλωματικής εργασίας είναι αξιόλογο αφού τα δίκτυα MobileNet Transfer Learning και MobileNet πετυχαίνουν σχεδόν πραγματικού χρόνου ανίχνευση και εντοπισμό οχημάτων. Επιπρόσθετα, τα ποσοστά ακριβείας εκπαίδευσης και επαλήθευσης όλων των εφαρμογών είναι εξαιρετικά (98%-100%). Προτείνει καινοτόμες λύσεις στην εργασία της παρακολούθησης της τροχαίας, της παρακολούθησης υπηρεσιακών οχημάτων κλπ ενώ οι λύσεις αυτές μπορούν να τροποποιηθούν ώστε να είναι χρήσιμες σε ιατρικές εφαρμογές (π.χ. πρόβλεψη ασθενειών μερικά χρόνια πριν συμβούν) στην περιοχή της Ψηφιακής Επεξεργασίας Εικόνας, της Μηχανικής Όρασης και της Μηχανικής Μάθησης.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Σκοπός και Σημασία της Παρούσας Ατομικής Διπλωματικής Εργασίας	
	1.2 Κίνητρο – Αιτιολόγηση του Θέματος	
	1.3 Δομή Ατομικής Διπλωματικής Εργασίας	
Κεφάλαιο 2	Ψηφιακή Επεξεργασία Εικόνας.....	20
	2.1 Ψηφιακή Επεξεργασία Εικόνας	
	2.2 Βασικά Χαρακτηριστικά Εικόνας	
	2.2.1 Pixels	
	2.2.2 Ανάλυση Εικόνας	
	2.2.3 Δημιουργία Εικόνας από Κανάλια Χρωμάτων	
	2.2.4 Σύστημα Συντεταγμένων Εικόνας	
	2.2.5 Εικόνες ως NumPy Πίνακες	
	2.2.6 Αλλαγή Μεγέθους Εικόνας και Αναλογία Απεικόνισης	
	2.3 Κατηγοριοποίηση Αντικειμένων στην Επεξεργασία Εικόνας	
	2.3.1 Γιατί χρησιμοποιούμε βαθιά συνελκτικά δίκτυα στην αναγνώριση αντικειμένων	
	2.3.2 Image Pyramid	
	2.4 Image Pyramid	
	2.5 Κατηγοριοποίηση Αντικειμένων στην Επεξεργασία Εικόνας	
	2.5.1 Image Classification	
	2.5.1.1 Image Classification Pipeline	
Κεφάλαιο 3	Μηχανική Όραση και Μηχανική Μάθηση	17
	3.1 Μηχανική Όραση	
	3.2 Τεχνητή Νοημοσύνη	
	3.2.1 Γνωστικά και Έμπειρα Συστήματα Μάθησης στην Τεχνητή Νοημοσύνη	
	3.2.2 Μηχανική Μάθηση	
	3.2.2.1 Τεχνητά Νευρωνικά Δίκτυα	
	3.2.2.1.1 Μάθηση	
	3.2.2.1.1.1 Κλάσεις Μάθησης:	
	3.2.2.1.1.1.1 Επιβλεπόμενη Μάθηση	

3.3.2.1.1.1.2	Ενισχυτική Μάθηση	
3.3.2.1.1.1.3	Μη Επιβλεπόμενη Μάθηση	
3.2.2.2	Τεχνητά Νευρωνικά Δίκτυα - Ορισμός και Ιστορική Αναδρομή	
3.2.2.3	Τεχνητά Νευρωνικά Δίκτυα – Δίκτυά και Σημαντικοί Αλγόριθμοι	
3.2.2.3.1	ΤΝΔ McCulloch and Pitts (McP)	
3.2.2.3.2	ΤΝΔ Perceptron	
3.2.2.3.3	Δίκτυα με ένα κρυφό Επίπεδο - Γραμμικά Διαχωρίσιμο Πρόβλημα	
3.2.2.3.4	Δίκτυο Πολυστρωματικών Επίπεδων Perceptron	
3.2.2.3.5	Αλγόριθμος Κατάβαση Κλίσης	
3.2.2.3.6	Αλγόριθμος Μηχανικής Μάθησης Perceptron	
3.2.2.3.7	Αλγόριθμος Μάθησης Ανάστροφης Μετάδοσης Σφάλματος	
3.2.2.4	Βαθιά Μάθηση	
3.2.2.4.1	Βαθιά Συνελικτικά Δίκτυα και Βαθιά Μάθηση - Ορισμός και Ιστορική Αναδρομή	
3.2.2.4.2	Βαθύ Συνελικτικό Δίκτυο	
3.2.2.4.3	Βαθιά Συνελικτικά Δίκτυα - Ιστορική Αναδρομή	
3.2.2.4.4	ImageNet	
3.2.2.4.5	Αλγόριθμοι και Optimizers Βαθιών Συνελικτικών Δικτύων που Χρησιμοποιήθηκαν	
3.2.2.4.5.1	Ρυθμός Μάθησης (Learning Rate)	
3.2.2.4.5.2	Ορμή	
3.2.2.4.5.3	Επιτάχυνση Nesterov	
3.2.2.4.5.4	Στοχαστική Κατάβαση Κλίσης	
3.2.2.4.5.5	RMSProp	
3.2.2.4.5.6	Adam	
3.2.2.4.5.7	Nadam	
3.2.2.4.5.8	Logistic Regression in Deep CNNs	
3.2.2.4.6	Αρχιτεκτονικές Βαθιών Συνελικτικών Δικτύων που Χρησιμοποιήθηκαν	
3.2.2.4.6.1	Επίπεδο Συνέλιξης.	
3.2.2.4.6.2	Επίπεδο Συγκέντρωσης	
3.2.2.4.6.3	Αρχιτεκτονική Συνελικτικού Δικτύου VGG16	
3.2.2.4.6.4	Αρχιτεκτονική Συνελικτικού Δικτύου ResNet50	
3.2.2.4.6.5	Αρχιτεκτονική Συνελικτικού Δικτύου ShallowNet	
3.2.2.4.6.6	Αρχιτεκτονική Συνελικτικού Δικτύου MobileNet	
3.2.2.5	Αναγνώριση Αντικειμένων vs Ανίχνευση Αντικειμένων vs Κατηγοριοποίηση στην Μηχανική Μάθηση και Μηχανική Όραση	

Κεφάλαιο 4	Βάσεις Δεδομένων και Επεξεργασία Δεδομένων.....	73
------------	---	----

4.1	Βάσεις Δεδομένων με Εικόνες Οχημάτων	
-----	--------------------------------------	--

4.2 Δεδομένα Εκπαίδευσης/Ελέγχου και Επαλήθευσης

4.3 Μορφή Δεδομένων

4.4 The Drone Dataset

4.4 Data Preprocessing

4.4.1 Data Augmentation

4.4.2 Aspect Aware Preprocessor

4.4.3 Image To Array Preprocessor

4.5 Μετρικές

4.6 Sliding Window

4.7 Non Maximum Suppression

4.8 Batch Preprocessing

4.9 Ensembles

Κεφάλαιο 5 Μεθοδολογία..... 80

5.1 Object Detection Pipeline

5.2 Μεθοδολογία Ατομικής Διπλωματικής Εργασίας

5.3 Χρήση Προεκπαιδευμένων Δικτύων Μάθησης

5.4 Λογισμικό που Χρησιμοποιήθηκε στην Υλοποίηση

5.4.1 OpenCV

5.4.2 TensorFlow και Keras

5.4.3 Scikit (Scikit Learn,..., etc)

5.4.4 NumPy

5.5 Transfer Learning

5.5.1 Transfer Learning, Fine-tuning και Εγχείρηση Συνελκτικού Δικτύου

5.6 Εξαγωγή Χαρακτηριστικών με την Χρήση Προ εκπαιδευμένων Δικτύων –

Παράδειγμα VGG16

5.6.1 Εξαγωγή Χαρακτηριστικών Οχημάτων –VGG16

5.6.2 Εξαγωγή Χαρακτηριστικών Οχημάτων – ResNet50

5.6.3 Εξαγωγή Χαρακτηριστικών Οχημάτων – ShallowNet

5.6.4 Εξαγωγή Χαρακτηριστικών Οχημάτων – MobileNet

5.6.5 Εξαγωγή Χαρακτηριστικών Οχημάτων– VGG16 finetuned

5.6.6 Εξαγωγή Χαρακτηριστικών Οχημάτων – MobileNet finetuned

5.7 Δημιουργία Δειγμάτων Εκπαίδευσης και Ελέγχου για DCNNs

5.7.1 Δημιουργία Δειγμάτων Εκπαίδευσης και Ελέγχου για Προεκπαιδευμένη

Έκδοση Συνελκτικού Δικτύου - VGG16

5.7.2 Δημιουργία Δειγμάτων Εκπαίδευσης και Ελέγχου για Προεκπαιδευμένη

Έκδοση Συνελκτικού Δικτύου - ResNet50

5.7.3 Δημιουργία Δειγμάτων Εκπαίδευσης και Ελέγχου για Προεκπαιδευμένη Έκδοση Συνελικτικού Δικτύου - ShallowNet

5.7.4 Δημιουργία Δειγμάτων Εκπαίδευσης και Ελέγχου για Προεκπαιδευμένη Έκδοση Συνελικτικού Δικτύου - MobileNet

5.7.6 Δημιουργία Δειγμάτων Εκπαίδευσης και Ελέγχου για Προεκπαιδευμένη fine-tuned Έκδοση Συνελικτικού Δικτύου - VGG16 finetuned

5.7.7 Δημιουργία Δειγμάτων Εκπαίδευσης και Ελέγχου για Προεκπαιδευμένη fine-tuned Έκδοση Συνελικτικού Δικτύου - MobileNet finetuned

5.8 Εγχείρηση Συνελικτικών Δικτύων

5.8.1 Εγχείρηση Προεκπαιδευμένων Εκδόσεων Συνελικτών Δικτύων

5.8.1.1 Εγχείρηση Προεκπαιδευμένης Έκδοσης Συνελικτικού Δικτύου - VGG16

5.8.1.2 Εγχείρηση Προεκπαιδευμένης Έκδοσης Συνελικτικού Δικτύου - ResNet50

5.8.1.3 Εγχείρηση Προεκπαιδευμένης Έκδοσης Συνελικτικού Δικτύου - ShallowNet

5.8.1.4 Εγχείρηση Προεκπαιδευμένης Έκδοσης Συνελικτικού Δικτύου - MobileNet

5.8.2 Εγχείρηση Προεκπαιδευμένων Εκδόσεων fine-tuned Συνελικτών Δικτύων

5.8.2.1 Εγχείρηση Προεκπαιδευμένης Έκδοσης fine-tuned Συνελικτικού Δικτύου - VGG16 fine-tuned

5.8.2.2 Εγχείρηση Προεκπαιδευμένης Έκδοσης fine-tuned Συνελικτικού Δικτύου - MobileNet fine-tuned

5.10 Ανίχνευσης Οχημάτων με την χρήση Συνελικτικών Δικτύων και Υλοποίηση της

5.10.1 Υλοποίηση Sliding Window

5.10.2 Υλοποίηση Image Pyramid

5.10.3 Batch Processing

5.10.4 Non maxima Suppression

Κεφάλαιο 6 Πειράματα και Αποτελέσματα..... 113

6.1 Πειράματα

6.1.1 Εξαγωγή Χαρακτηριστικών Οχημάτων

6.1.1.1. ResNet50

6.1.1.2 VGG16

6.1.1.3 ShallowNet

6.1.1.4 MobileNet

6.1.1.5 VGG16 Transfer Learning

6.1.1.6 MobileNet Transfer Learning

6.1.2 Εκπαίδευση, Έλεγχος και Επικύρωση Μοντέλων Αναγνώρισης και Ανίχνευσης Οχημάτων

6.1.2.1 Επιλογή Αλγορίθμων Ενημέρωσης Συναπτικών Βαρών και των

Παραμέτρων τους

6.1.2.1.1 ResNet50

6.1.2.1.2 VGG16

6.1.2.1.3 ShallowNet

6.1.2.1.4 MobileNet

6.1.2.1.5 VGG16 Transfer Learning

6.1.2.1.6 MobileNet Transfer Learning

6.1.3 Εκπαίδευση και Έλεγχος Ανιχνευτών Οχημάτων

6.1.3.1 Επιλογή Αλγορίθμων Ενημέρωσης Συναπτικών Βαρών και των

Παραμέτρων τους

6.1.3.1.1 ResNet50

6.1.3.1.2 VGG16

6.1.3.1.3 ShallowNet

6.1.3.1.4 MobileNet

6.1.3.1.5 VGG16 Transfer Learning

6.1.3.6.1 MobileNet Transfer Learning

6.2 Αποτελέσματα

6.2.1 Αποτελέσματα Εξαγωγής Χαρακτηριστικών Οχημάτων

6.2.1.1. ResNet50

6.2.1.2 VGG16

6.2.1.3 ShallowNet

6.2.1.4 MobileNet

6.2.1.5 VGG16 Transfer Learning

6.2.1.6 MobileNet Transfer Learning

6.2.2 Εκπαίδευση και Έλεγχος Ανιχνευτών Οχημάτων

6.2.2.1 ResNet50

6.2.2.2 VGG16

6.2.2.3 ShallowNet

6.2.2.4 MobileNet

6.2.2.5 VGG16 Transfer Learning

6.2.2.6 MobileNet Transfer Learning

6.2.3 Αποτελέσματα Ανιχνευτών Οχημάτων

6.2.3.1 Αποτελέσματα Simple Detection Drone ResNet50 Logistic Regression

6.2.3.2 Αποτελέσματα Simple Detection Drone VGG16 Logistic Regression

6.2.3.3 Αποτελέσματα Simple Detection Drone ShallowNet Logistic Regression

6.2.3.4 Αποτελέσματα Simple Detection Drone MobileNet Logistic Regression

6.2.3.5 Αποτελέσματα Simple Detection Drone VGG16 Logistic Regression

6.2.3.6 Αποτελέσματα Simple Detection Drone MobileNet Transfer Learning

Logistic Regression

ΚΕΦΑΛΑΙΟ 7 Συμπεράσματα και Μελλοντική Εργασία 135

7.1 Συμπεράσματα

7.2 Μελλοντική Εργασία

ΠΑΡΑΡΤΗΜΑ Α Εντολές Εκτέλεσης Εφαρμογών

ΠΑΡΑΡΤΗΜΑ Β Τιμές που Δοκιμάστηκαν στους Ανιχνευτές Οχημάτων

ΠΑΡΑΡΤΗΜΑ Γ - Κώδικας

Κεφάλαιο 1

Εισαγωγή

1.1 Σκοπός και Σημασία της Παρούσας Ατομικής Διπλωματικής Εργασίας	1
1.2 Κίνητρο – Αιτιολόγηση του Θέματος	3
1.3 Δομή Ατομικής Διπλωματικής Εργασίας	4

1.1 Σημασία και Σκοπός της Παρούσας Ατομικής Διπλωματικής Εργασίας

Η μηχανική όραση είναι η επιστήμη που βοηθά τις μηχανές βλέπουν. Η επιστήμη αυτή ασχολείται με την θεωρία των συστημάτων τεχνητής νοημοσύνης που εξάγουν πληροφορίες από εικόνες. Οι εικόνες εισόδου μπορούν να έχουν οποιαδήποτε μορφή όπως εικόνες, ακολουθίες εικόνων από βίντεο, όψεις από διάφορες κάμερες, πολυδιάστατα δεδομένα από ιατρικούς σαρωτές (π.χ. αξονικός τομογράφος CAT, μαγνητικός τομογράφος MRI). Η μηχανική όραση είναι άμεσα συνδεδεμένη με την βιολογική όραση. Το πεδίο της (βιολογικής) όρασης μελετά και μοντελοποιεί τις φυσικές διεργασίες πίσω από την οπτική αντίληψη των ανθρώπων και άλλων ζώων. Από την άλλη η μηχανική όραση περιγράφει τις διεργασίες πίσω από τα τεχνητά συστήματα όρασης που υλοποιούνται με λογισμικό και υλικό. Η διεπιστημονική ανταλλαγή ιδεών είναι ιδιαίτερα καρποφόρα μεταξύ των δύο πιο πάνω επιστημονικών πεδίων [9].

Το κλασικό πρόβλημα που προσπαθούν να επιλύσουν η ψηφιακή επεξεργασία εικόνας και η μηχανική όραση είναι η απόφαση αν μέσα σε μια εικόνα (δεδομένο εισόδου) υπάρχει ένα συγκεκριμένο αντικείμενο ή χαρακτηριστικό. Αυτή η εργασία μπορεί να επιλυθεί εύκολα χωρίς την επέμβαση κάποιου ανθρώπου, παρόλο που δεν έχει επιλυθεί στην γενική περίπτωση κατά πόσο οποιοδήποτε αντικείμενο βρίσκεται σε οποιαδήποτε εικόνα στο πεδίο της μηχανικής όρασης [9].

Ένα ενδιαφέρον πρόβλημα με το οποίο ασχολείται η μηχανική όραση, η ψηφιακή επεξεργασία εικόνας και η μηχανική μάθηση αυτή την περίοδο είναι η ανίχνευση και αναγνώριση των ανθρώπινων προσώπων. Η αναγνώριση και η ανίχνευση προσώπων βρίσκει εφαρμογές στην ταυτοποίηση ατόμων, στην αναγνώριση προσώπων (ετικετοποίηση τους) σε φωτογραφίες στα μέσα κοινωνικής δικτύωσης, στην ετικετοποίηση εικόνων κ.λπ.

Ένα εξίσου ενδιαφέρον πρόβλημα είναι η αναγνώριση, ο εντοπισμός και η ανίχνευση οχημάτων. Το πρόβλημα αυτό είναι σημαντική και βασική εργασία στην παρακολούθηση της τροχιάς κίνησης και σε άλλες εφαρμογές όπως η ανίχνευση και ο εντοπισμός πυροσβεστικών και αστυνομικών οχημάτων που βρίσκονται σε υπηρεσία, λεωφορείων που εκτελούν κάποιο δρομολόγιο κ.ά.

Τα βαθιά συνελκτικά δίκτυα είναι νευρωνικά δίκτυα εμπνευσμένα από την οργάνωση του οπτικού εγκεφαλικού φλοιού των ζώων. Μεμονωμένοι νευρώνες του ματιού ανταποκρίνονται σε ερεθίσματα σε μια περιορισμένη περιοχή του χώρου (receptive field). Τα receptive fields διαφορετικών νευρώνων επικαλύπτονται δημιουργώντας το οπτικό πεδίο. Η ανταπόκριση ενός μεμονωμένου νευρώνα σε ερεθίσματα μέσα στο receptive field μπορεί να προσεγγιστεί μαθηματικά με μια λειτουργία συνέλιξης. Τα βαθιά συνελκτικά δίκτυα χρησιμοποιούνται σε εφαρμογές που τα δεδομένα εισόδου είναι σε μορφή εικόνας, ήχου ή βίντεο. Τα δίκτυα αυτά παίρνουν ως είσοδο χαρακτηριστικά των δεδομένων εισόδου και εκτελούν μια συγκεκριμένη εργασία σε αυτά (κατηγοριοποίηση ή πρόβλεψη) π.χ αναγνώριση ενός συγκεκριμένου αντικειμένου σε μια εικόνα ή ένα βίντεο, αναγνώριση ενός ηχητικού μοτίβου σε ένα τραγούδι κ.λ.π [1, 16].

Τα μη επανδρωμένα εναέρια μέσα (drones) είναι μικρές ιπτάμενες συσκευές που έχουν μια κάμερα τοποθετημένη πάνω τους. Σε αυτά δεν επιβιβάζεται πλήρωμα και γι' αυτό ονομάζονται μη επανδρωμένα εναέρια μέσα. Τα drones πετούν με την χρήση τηλεχειριστηρίου. Ένας χειριστής (άνθρωπος) πετάει το drone με την χρήση συγκεκριμένων πλήκτρων που βρίσκονται στο τηλεχειριστήριο. Το drone έχει ενσωματωμένη μια κάμερα από την κατασκευή του με την οποία βιντεοσκοπεί το τοπίο πάνω από το οποίο πετά. Με τη χρήση του drone μπορούν να ληφθούν φωτογραφίες και βίντεο [9].

Το θέμα της παρούσας διπλωματικής εργασίας είναι η αναγνώριση, η ανίχνευση και ο εντοπισμός των οχημάτων που βρίσκονται στις εικόνες που λήφθηκαν από μη επανδρωμένα εναέρια μέσα (drones) με την χρήση βαθιών συνελικτικών δικτύων.

Στην παρούσα διπλωματική εργασία προσπαθούμε να αναγνωρίσουμε, να ανιχνεύσουμε και να εντοπίσουμε οχήματα που κινούνται μέσα σε δρόμους με τη χρήση πολλών διαφορετικών αρχιτεκτονικών συνελικτικών δικτύων καθώς και των βελτιστοποιημένων εκδόσεων αυτών.

Τα δεδομένα εισόδου στην εφαρμογή μας χωρίζονται σε δύο κατηγορίες. Η πρώτη κατηγορία εικόνων είναι εικόνες με άδειους δρόμους και η δεύτερη είναι εικόνες με δρόμους στους οποίους υπάρχουν αυτοκίνητα. Στη συνέχεια, γίνεται εξαγωγή των χαρακτηριστικών των εικόνων των δύο κατηγοριών.

Ακολούθως, τα βαθιά συνελικτικά δίκτυα παίρνουν ως είσοδο τα χαρακτηριστικά του συνόλου δεδομένων μας και εκπαιδεύονται χρησιμοποιώντας προ εκπαιδευμένα βάρη. Κατά την εκπαίδευση, κατηγοριοποιούμε μια εικόνα σύμφωνα με το αν σε αυτή υπάρχουν οχημάτων ή όχι (αναγνώριση οχημάτων). Επιπρόσθετα, διεξάγουμε τον έλεγχο και την επαλήθευση των συνελικτικών δικτύων που εκπαιδεύσαμε για να δούμε κατά πόσο μπορεί το δίκτυο μας να αναγνωρίσει σε νέες εικόνες, που δεν έχει ξαναδεί κατά την εκπαίδευση, αν περιέχει κάποιο αυτοκίνητο ή όχι. Με βάσει τις πιο πάνω κατηγοριοποιήσεις, κατασκευάζουμε ένα μοντέλο το οποίο θα μας βοηθήσει αργότερα στην ανίχνευση και στον εντοπισμό των οχημάτων που βρίσκονται μέσα σε εικόνες.

Στη συνέχεια, ένας ανιχνευτής αντικειμένων παίρνει ως είσοδο το μοντέλο που κατασκευάσαμε κατά την εκπαίδευση και ψάχνει να βρει τα οχήματα που βρίσκονται μέσα στην εικόνα που του δώσαμε ως δεδομένο εισόδου. Τέλος, ο ανιχνευτής αυτός μπορεί να εντοπίσει περισσότερα από ένα αυτοκίνητα που βρίσκονται μέσα σε μια εικόνα και να προσδιορίσει την ακριβή τους θέση δημιουργώντας ένα κουτί γύρο από κάθε αυτοκίνητο

που εντοπίζει. Η πιο πάνω διαδικασία ακολουθήθηκε για την εκπαίδευση, τον έλεγχο και την επαλήθευση πολλών και διαφορετικών βαθιών συνελκτικών δικτύων και βελτιστοποιημένων εκδόσεων τους καθώς και για την εκπαίδευση και τον έλεγχο των διαφορετικών ανιχνευτών και των βελτιστοποιημένων εκδόσεών αυτών.

1.2 Κίνητρο – Αιτιολόγηση του Θέματος

Η αναγνώριση και ανίχνευση αντικειμένων είναι ένα ενδιαφέρον πρόβλημα στην μηχανική όραση, την μηχανική μάθηση και την ψηφιακή επεξεργασία εικόνας. Η γενική λύση του προβλήματος θα δώσει την δυνατότητα για ανάπτυξη ποικίλων και χρήσιμων εφαρμογών οι οποίες θα μειώνουν τον χρόνο που δαπανείται για αναζήτηση αντικειμένων σε εικόνες και βίντεο, θα αυξήσουν τα ποσοστά ακριβείας στην σωστή ανίχνευση και αναγνώριση αντικειμένων σε ακολουθίες εικόνων και θα βελτιώσουν την ποιότητα ζωής μας.

Στην παρούσα διπλωματική εργασία αποφασίσαμε να ασχοληθούμε με το πρόβλημα της ανίχνευσης και αναγνώρισης οχημάτων σε εικόνες διότι αυτό αποτελεί βασική εργασία στην παρακολούθηση της τροχιάς κίνησης και σε άλλες εφαρμογές που υπάρχουν σε έξυπνες πόλεις. Μια χρήσιμη εφαρμογή είναι ο πραγματικού χρόνου εντοπισμός οχημάτων σε βίντεο. Αυτή η εφαρμογή μπορεί να χρησιμοποιηθεί από τα έξυπνα αυτοκίνητα (που κινούνται χωρίς οδηγό) ώστε να αποφύγουν πιθανά τροχαία ατυχήματα.

Υπάρχουν δύο τρόποι με τους οποίους μπορεί να γίνει η αναγνώριση και η ανίχνευση αντικειμένων στους τομείς της μηχανικής μάθησης και όρασης και ψηφιακής επεξεργασίας εικόνας. Ο πρώτος τρόπος είναι με την χρήση κλασσικών μεθόδων της μηχανικής όρασης και μάθησης για ανίχνευση αντικειμένων. Αυτές περιλαμβάνουν μεθόδους όπως τα κινητά παράθυρα στον χρόνο και οι πυραμίδες εικόνων (οι μέθοδοι που χρησιμοποιούνται στους HOG και linear SVM ανιχνευτές αντικειμένων). Ο δεύτερος τρόπος είναι με την χρήση προεκπαιδευμένων δικτύων και η χρήση τους ως βασικό δίκτυο μια ευρύτερη αρχιτεκτονική δικτύου (για παράδειγμα Fast R-CNN, SSD, YOLO) [11].

Ακόμη ένας λόγος που μας ώθησε στην επιλογή του συγκεκριμένου θέματος είναι η ύπαρξη μιας μεταπτυχιακής ερευνητικής εργασίας που το αντικείμενο της ήταν η ανίχνευση και η ιχνηλάτηση οχημάτων με την χρήση κλασσικών μεθόδων μηχανικής μάθησης και μηχανικής όρασης. Στην παρούσα διπλωματική εργασία μελετούμε την κατηγοριοποίηση, αναγνώριση και ανίχνευση οχημάτων με την χρήση βαθιών συνελικτικών δικτύων συμπληρώνοντας τους δύο τρόπους με τους οποίους γίνεται η αναγνώριση και ανίχνευση αντικειμένων στην μηχανική όραση.

Ένας άλλος λόγος που μας οδήγησε στην επιλογή του παρόν θέματος και την ενασχόληση με αυτό είναι η ύπαρξη των βαθιών συνελικτικών δικτύων και οι εξαιρετικές τους επιδόσεις σε διαγωνισμούς οπτικής αναγνώρισης αντικειμένων.

Ένας επιπλέον λόγος που μας οδήγησε στην επιλογή και ερευνητική ενασχόληση με το παρόν θέμα είναι ενός συνόλου δεδομένων που περιλάμβανε οχήματα και άδειους δρόμους. Το προαναφερθέν σύνολο δεδομένων δημιουργήθηκε με την χρήση μη επανδρωμένων εναέριων μέσων (drones) και κατασκευάστηκε για να χρησιμοποιηθεί στην εκπόνηση της μεταπτυχιακής ερευνητικής εργασίας με το αντικείμενο την ανίχνευση και η ιχνηλάτηση οχημάτων χρησιμοποιώντας κλασσικές μεθόδους μηχανικής μάθησης και μηχανικής όρασης (HOG & Linear SVM classifiers) [9].

Τέλος, ένας άλλος λόγος επιλογής αυτού του θέματος είναι το προσωπικό μου ενδιαφέρον για τα βαθιά συνελικτικά δίκτυα. Συγκεκριμένα, έχω παρακολουθήσει διαδικτυακά μαθήματα από την NVIDIA που αφορούν τα βαθιά συνελικτικά δίκτυα (2015) και εργαλεία όπως το Keras και το Tensorflow από την IBM (2018) και πως αυτά χρησιμοποιούνται για την εκπαίδευση, τον έλεγχο και την επαλήθευση βαθιών συνελικτικών δικτύων. Τον Μάρτιο του 2016 είχα την τύχη να παρακολουθήσω μέσω ζωντανής μετάδοσης (live stream) την τελευταία μισή ώρα του τελευταίου αγώνα AlphaGo

μεταξύ του παγκόσμιου πρωταθλητή στο AlphaGo και του βαθιού συνελικτικού δικτύου που δημιούργησε η DeepMind.

1.3 Δομή Ατομικής Διπλωματικής Εργασίας

Η ατομική διπλωματική εργασία είναι οργανωμένη στα πιο κάτω κεφάλαια:

Κεφάλαιο 2: Μηχανική Όραση και Μηχανική Μάθηση Σε αυτό το κεφάλαιο παρέρχεται το θεωρητικό υπόβαθρο της μηχανικής όρασης και μάθησης όπως τα νευρωνικά δίκτυα, οι αλγόριθμοι και η αρχιτεκτονική των δικτύων που θα χρησιμοποιηθούν.

Κεφάλαιο 3: Ψηφιακή Επεξεργασία Εικόνας. Στο κεφάλαιο αυτό παρέρχονται βασικές έννοιες της ψηφιακής επεξεργασίας εικόνας όπως τα βασικά χαρακτηριστικά μιας εικόνας και έννοιες που αφορούν την κατηγοριοποίηση αντικειμένων.

Κεφάλαιο 4 Βάσεις Δεδομένων και Επεξεργασία Δεδομένων. Στο τέταρτο κεφάλαιο παρουσιάζονται οι βάσεις δεδομένων που υπάρχουν στο διαδίκτυο και οι οποίες σχετίζονται με την ανίχνευση οχημάτων. Επίσης γίνεται παρουσίαση της βάσης δεδομένων με εναέριες λήψεις οχημάτων που χρησιμοποιήθηκε για τους σκοπούς της συγκεκριμένης ατομικής διπλωματικής εργασίας καθώς και τεχνικές προ επεξεργασίας δεδομένων που χρησιμοποιήθηκαν στα πλαίσια αυτής. Τέλος, γίνεται αναφορά σε διάφορες τεχνικές που χρησιμοποιούν τα βαθιά συνελικτικά δίκτυα και χρησιμοποιούνται στην παρούσα διπλωματική εργασία.

Κεφάλαιο 5 Μεθοδολογία. Στο συγκεκριμένο κεφάλαιο αναλύεται η διαδικασία που ακολουθήθηκε για την κατασκευή των μοντέλων αναγνώρισης, ανίχνευσης οχημάτων (η εκπαίδευση, έλεγχος, επαλήθευση των διαφόρων βαθιών συνελικτικών δικτύων) καθώς και η διαδικασία αναγνώρισης και ανίχνευσης οχημάτων (εκπαίδευση των διαφόρων ανιχνευτών οχημάτων). Ακόμη, γίνεται αναφορά και στα εργαλεία που χρησιμοποιήθηκαν

για την δημιουργία των αλγορίθμων αναγνώρισης και ανίχνευσης των οχημάτων καθώς και οι ίδιοι αλγόριθμοι που χρησιμοποιήθηκαν.

Κεφάλαιο 6 Πειράματα και Αποτελέσματα. Στο έκτο κεφάλαιο παρουσιάζεται το 1/5 των πειραμάτων που έγιναν κατά την εκπόνηση αυτής τις διπλωματικής εργασίας και τα αποτελέσματα που σχετίζονται με εξαγωγή των χαρακτηριστικών ενός οχήματος, την κατασκευή των μοντέλων αναγνώρισης και ανίχνευσης οχημάτων καθώς και με την ανίχνευση οχημάτων καθαυτή.

Κεφάλαιο 7 Συμπεράσματα και Μελλοντική Εργασία. Στο τελευταίο κεφάλαιο αναφέρονται τα συμπεράσματα της διπλωματικής εργασίας και προτείνονται εισηγήσεις για μελλοντική επέκταση της παρούσας διπλωματικής εργασίας.

Κεφάλαιο 2

Ψηφιακή Επεξεργασία Εικόνας

2.1 Ψηφιακή Επεξεργασία Εικόνας	1
2.2 Βασικά Χαρακτηριστικά Εικόνας	3
2.3 Κατηγοριοποίηση Αντικειμένων στην Επεξεργασία Εικόνας	4

2.1 Ψηφιακή Επεξεργασία Εικόνας

Η εικόνα είναι μια συνάρτηση $I(w, h)$ όπου το I είναι η ένταση της φωτεινότητας της. Το δομικό στοιχείο μιας εικόνας είναι το pixel. Κάθε pixel περιγράφεται με τρεις τιμές. Οι δύο τιμές προσδιορίζουν την θέση του pixel πάνω στην εικόνα και η τρίτη τιμή καθορίζει την φωτεινότητά του. Η συνάρτηση $I(w, h)$ δίνει την τιμή της φωτεινότητας του pixel στη συγκεκριμένη θέση (w, h) της εικόνας αφού η εικόνα ορίζεται ως ένας πίνακας πεπερασμένων διαστάσεων. Μια εικόνα ονομάζεται ψηφιακή όταν οι τιμές των διαστάσεων της (w, h) και της έντασης I της είναι πεπερασμένες και διακριτές [9].

2.2 Βασικά Χαρακτηριστικά Εικόνας

2.2.1 Pixels

Το δομικό υλικό μιας εικόνας είναι το pixel. Κάθε εικόνα απαρτίζεται από ένα σύνολο pixels. Το pixel δεν μπορεί να διασπαστεί σε πιο μικρά κομμάτια. Ένα pixel θεωρείται το χρώμα ή το πόσο έντονα φαίνεται το φως σε ένα συγκεκριμένο σημείο μέσα στην εικόνα μας [9, 12].

Μπορούμε να φανταστούμε την εικόνα ως ένα πολυδιάστατο πίνακα. Στο παράδειγμά μας, ο πίνακας μας έχει 1000 στήλες (πλάτος) και 750 γραμμές (μήκος). Το συνολικό πλήθος των pixels που βρίσκονται στην εικόνα μας είναι $1000 \times 750 = 750\,000$ pixels [12].

Τα pixels αναπαρίστανται με δύο τρόπους. Ο πρώτος τρόπος είναι grayscale/ένα κανάλι και ο δεύτερος τα έγχρωμα. Σε μια grayscale εικόνα κάθε pixel είναι μια κλιμακωτή τιμή μεταξύ των τιμών 0 και 255 συμπεριλαμβανομένων. Οι τιμές αυτές, δείχνουν τις αποχρώσεις του γκρι που μπορούν να υπάρξουν σε μια grayscale εικόνα. Συγκεκριμένα, η

τιμή 0 δείχνει ότι το pixel είναι μαύρο ενώ η τιμή 255 ότι είναι άσπρο. Όσο πιο κοντά στο 0 βρίσκεται η τιμή τόσο πιο σκουρόχρωμη είναι η απόχρωση του γκρι. Όσο πιο κοντά στο 255 βρίσκεται η τιμή τόσο πιο ανοιχτόχρωμη είναι η απόχρωση του γκρι [12].

Τα pixels που βρίσκονται στον RGB χώρο χρωμάτων δεν αντιπροσωπεύονται με κλιμακωτές τιμές. Αντίθετα, τα pixels αυτά αναπαρίστανται ως μια λίστα με τρεις τιμές, μια τιμή για το κόκκινο χρώμα, μια τιμή για το πράσινο χρώμα και μια τιμή για το μπλε. Μπορούμε να ορίσουμε ένα χρώμα στον RGB χώρο χρωμάτων καθορίζοντας την ποσότητα του κόκκινου, πράσινου και μπλε χρώματος που θα περιέχει ένα pixel. Κάθε κανάλι ενός χρώματος (Κόκκινο, Πράσινο, Μπλε) περιέχει τιμές μεταξύ 0 και 255 συμπεριλαμβανομένων (συνολικά 256 σκιάς). Η τιμή 0 δείχνει ότι το pixel δεν αναπαρίσταται (μαύρο χρώμα) ενώ το 255 πώς αναπαρίσταται (άσπρο χρώμα). Συνήθως, χρησιμοποιούμε ακέραιους απρόσημους αριθμούς μεγέθους 8 bits για να αναπαραστήσουμε το πόσο έντονο είναι ένα χρώμα [12].

2.2.2 Ανάλυση Εικόνας

Ο όρος αυτός αναφέρεται στο συνολικό πλήθος των pixels που προκύπτει από το γινόμενο του πλάτους W (πλήθος των pixels στις στήλες) με το μήκος H (πλήθος των pixels στις γραμμές) Η ανάλυση της εικόνας διατηρεί ανάλογη σχέση με την ποιότητα αυτής αλλά και με το μέγεθος S της μνήμης που καταλαμβάνει [9].

2.2.3 Δημιουργία Εικόνας από Κανάλια Χρωμάτων

Όπως γνωρίζουμε μια RGB εικόνα αναπαρίσταται με τρεις τιμές, μια για το κόκκινο, μια για το πράσινο και μια για το μπλε στοιχείο μιας εικόνας αντίστοιχα. Μπορούμε να φανταστούμε πως μια RGB εικόνα αποτελείται από τρεις ανεξάρτητους πίνακες πλάτους

W και μήκους H , έναν για κάθε RGB στοιχείο. Στον χώρο χρωμάτων RGB, κάθε κανάλι ενός χρώματος είναι ένας ξεχωριστός πίνακας που όταν συνδυαστεί με τους υπόλοιπους πίνακες χρωμάτων του χώρου RGB, δημιουργώντας μια έγχρωμη εικόνα $W \times H \times D$ όπου το D είναι το βάθος ή το πλήθος των καναλιών των χρωμάτων. Στις εφαρμογές που υλοποιήσαμε, κάθε κανάλι χρωμάτων αναπαρίσταται με τιμή μεταξύ $[0, 255]$. Ενώ ένα pixel σε μια εικόνα RGB αναπαρίσταται ως μια λίστα από τρεις ακέραιες τιμές, μια για το κόκκινο, δεύτερη για το πράσινο και Τρίτη για το μπλε χρώμα [12]. Στις εφαρμογές της παρούσας διπλωματικής εργασίας, μια εικόνα αναπαρίσταται προγραμματιστικά ως ένας τρισδιάστατος πίνακας NumPy με πλάτος, μήκος και βάθος.

2.2.4 Σύστημα Συντεταγμένων Εικόνας

Μια εικόνα αναπαρίσταται ως ένα πλέγμα από pixels. Το σημείο έναρξης της εικόνας $(0,0)$ αντιστοιχεί στην πάνω αριστερή γωνία της εικόνας. Ακολουθώντας, όσο πιο δεξιά και κάτω κατευθυνόμαστε στην εικόνα παρατηρούμε πως οι τιμές x και y αυξάνονται. Θυμίζουμε πως η γλώσσα προγραμματισμού που χρησιμοποιούμε για την εφαρμογή μας είναι η python, στην οποία οι δείκτες ξεκινούν να παίρνουν τιμές από 0, γι' αυτό και το πρώτο σημείο μιας εικόνας είναι το $(0,0)$ [12].

2.2.5 Εικόνες ως NumPy Πίνακες

Βιβλιοθήκες ψηφιακής επεξεργασίας εικόνας όπως οι OpenCV και scikit image αναπαριστούν τις εικόνες RGB ως πολυδιάστατους NumPy πίνακες που έχουν σχήμα (height, width, depth). Το σχήμα των πινάκων με τον πιο πάνω τρόπο επειδή όταν δηλώνουμε τις διαστάσεις ενός πίνακα γράφουμε τις διαστάσεις του ως γραμμές επί

στήλες ,οι γραμμές είναι το height και οι στήλες το width της εικόνας, το βάθος του πίνακα είναι το με βάθος της εικόνας [12].

2.2.6 Αλλαγή Μεγέθους Εικόνας και Αναλογία Απεικόνισης

Η αλλαγή μεγέθους μιας εικόνας είναι μια διαδικασία μέσω της οποίας μειώνουμε ή αυξάνουμε το μέγεθος μιας εικόνας όσον αφορά το πλάτος και το μήκος της. Κατά την αλλαγή του μεγέθους μιας εικόνας, χρειάζεται να γνωρίζουμε την αναλογία απεικόνισης της και να την διατηρήσουμε. Η αναλογία απεικόνισης είναι η αναλογία του πλάτους ως προς το μήκος μιας εικόνας. Αν αγνοήσουμε την αναλογία αυτή οι εικόνες θα φαίνονται παραμορφωμένες. Για να αποφύγουμε αυτό το φαινόμενο αλλάζουμε το μέγεθος μιας εικόνας αυξάνοντας ή μειώνοντας το μήκος και το πλάτος της με την ίδια τιμή. Οι συνήθεις αναλογίες εικόνων είναι 1.33:1 και 1.77:1 [9, 12].

Στα τεχνητά νευρωνικά δίκτυα και ειδικότερα στα συνελκτικά δίκτυα οι εικόνες χρειάζεται να έχουν ένα συγκεκριμένο μέγεθος. Με αυτό εννοούμε πως οι διαστάσεις όλων των εικόνων που θα δοθούν σε ένα συγκεκριμένο δίκτυο πρέπει να είναι οι ίδιες. Θυμίζουμε πως θα γίνει η κατάλληλη προεπεξεργασία των εικόνων του συνόλου δεδομένων μας πριν αυτό δοθεί στα δίκτυα. Μια προεπεξεργασία εικόνων που θα πραγματοποιήσουμε είναι η mean subtraction ή scaling που απαιτεί την μετατροπή των εικόνων σε τύπο δεδομένων floating point. Αυτή η μετατροπή και παρόμοιες μετατροπές θα μας φανούν ιδιαίτερα χρήσιμες κατά την χρήση των βιβλιοθηκών που (πχ OpenCV) διότι αυτές χρησιμοποιούν διαφορετικούς τύπους δεδομένων τους οποίους πρέπει αργότερα να μετατρέψουμε σε άλλους για να μπορούμε να εφαρμόσουμε αλγορίθμους μάθησης σε αυτούς [12].

Χαρακτηριστικά (Features)

Στον τομέα της επεξεργασίας εικόνας, της μηχανικής μάθησης και της μηχανικής όρασης, ο όρος χαρακτηριστικά αναφέρεται τα σημεία του αντικειμένου που μας ενδιαφέρει και μπορούν να βρεθούν με την χρήση μιας αξιόπιστης τεχνικής εντοπισμού. Μια αξιόπιστη τεχνική εντοπισμού, είναι αξιόπιστη, ως προς την ακρίβεια εντοπισμού των σημείων του αντικειμένου ενδιαφέροντος. Η αξιопιστία της προαναφερθείσας τεχνικής πηγάζει από το γεγονός πως τα ίδια σημεία που αφορούν το ίδιο αντικείμενο ενδιαφέροντος, εντοπίζονται σε διαφορετικές εικόνες. Τέλος, τα χαρακτηριστικά χρησιμοποιούνται σε πολλές εφαρμογές (π.χ στην ανίχνευση αντικειμένων, στην ανακατασκευή του χώρου κλπ) [9].

2.3 Κατηγοριοποίηση Αντικειμένων στην Επεξεργασία Εικόνας

Από το 2005, σημειώθηκαν αρκετές επιτυχίες στην ανίχνευση, κατάτμηση, αναγνώριση αντικειμένων και περιοχών σε εικόνες. Στο πιο πάνω αποτέλεσμα συντέλεσε το πλήθος των ετικετοποιημένων δεδομένων (π.χ. σήματα κώδικα οδικής κυκλοφορίας, πεζοί, πρόσωπα) που υπάρχει στο διαδίκτυο [9].

Το 2012, στον διαγωνισμό ImageNet συνέβη ένα πρωτόγνωρο γεγονός. Ένα σύνολο δεδομένων που αποτελείτο από 1 million εικόνες κατηγοριοποιήθηκε σύμφωνα με 1000 κλάσεις με το μισό ποσοστό σφάλματος σε σχέση με αντίστοιχες προσπάθειες που έγιναν στο παρελθόν [9].

Τα βαθιά συνελκτικα δίκτυα πλέον αποτελούν την συνήθη πρακτική στις εργασίες οπτικής αναγνώρισης και χρησιμοποιούνται από πάρα πολλές εταιρίες-κολοσσούς όπως η Google, η Microsoft, η IBM αλλά και από πολλές νεοφυείς επιχειρήσεις [1].

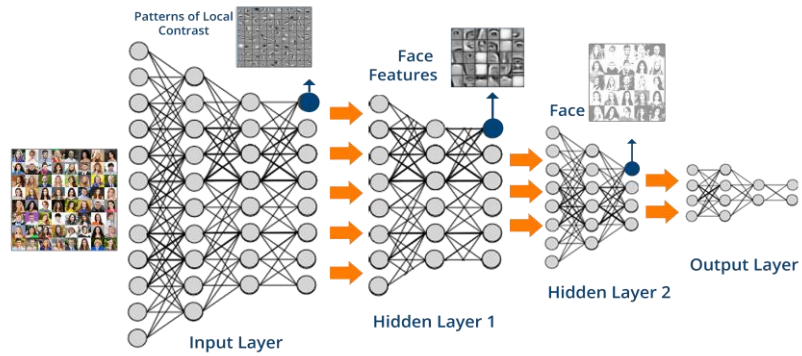
Οι εταιρείες NVIDIA, MobileEye, Intel, Qualcomm και Samsung αναπτύσσουν chips βαθιών συνελκτικών δικτύων που βοηθούν στην υλοποίηση πραγματικού χρόνου εφαρμογών μηχανικής όρασης σε κινητά τηλέφωνα, κάμερες, ρομποτς και αυτοοδηγούμενα

αυτοκίνητα. Στο παρελθόν (1990), αναπτύχθηκαν chips νευρωνικών δικτύων από την Siemens για να βοηθήσουν στην υλοποίηση πραγματικού χρόνου εφαρμογών μηχανικής μάθησης [16].

Στο πεδίο αυτό υπάρχουν πολλές προκλήσεις όπως η επιλογή της κατάλληλης αρχιτεκτονικής βαθιού συνελκτικού νευρωνικού δικτύου για επίλυση ενός προβλήματος, ο μεγάλος όγκος δεδομένων που χρειάζονται τα βαθιά συνελκτικά δίκτυα για να εκπαιδευτούν, η ανακάλυψη τεχνικών παραγωγής περισσότερων δεδομένων εκπαίδευσης με την αποσύνθεση υπάρχοντων παραδειγμάτων [10,16].

2.3.1 Γιατί Χρησιμοποιούμε Βαθιά Συνελκτικά Δίκτυα στην Αναγνώριση Αντικειμένων

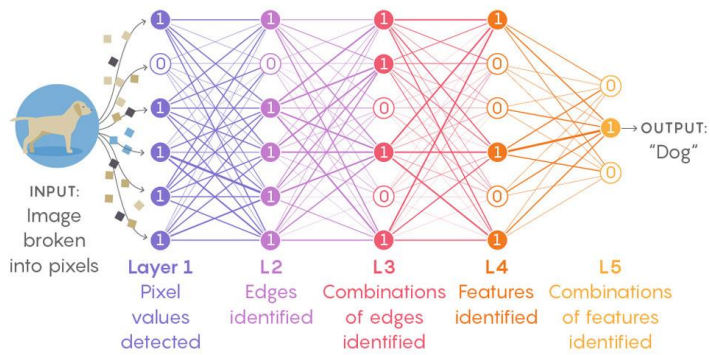
Η χρήση των βαθιών συνελκτικών δικτύων είναι απαραίτητη στην αναγνώριση αντικειμένων μέσα σε μια εικόνα. Οι εικόνες περιέχουν πολλές περιττές πληροφορίες ενώ παράλληλα η σημαντικότερη πληροφορία που χρειαζόμαστε στην αναγνώριση αντικειμένων είναι οι ακμές. Επομένως, σύμφωνα με αυτή τη γνώση νευρωνικό, χρησιμοποιούμε ένα μηχανισμό συνέλιξης όπως αυτού που υπάρχει στα συνελκτικά δίκτυα για να μας βοηθήσει να ανιχνεύσουμε τις ακμές. Ο κύριος λόγος που χρησιμοποιούμε συνελκτικά νευρωνικά δίκτυα στις εργασίες μάθησης μηχανικής όραση είναι πως τα δίκτυα αυτά χρησιμοποιούν τοπικά πρότυπα δεδομένων και η αρχιτεκτονική τους είναι ειδικά σχεδιασμένη για επίλυση προβλημάτων αναγνώρισης και ανίχνευσης αντικειμένων σε εικόνες ή βίντεο. Ακόμη ένας λόγος που μας ώθησε στην χρήση βαθιών συνελκτικών δικτύων όσον αφορά τις εφαρμογές αναγνώρισης και ανίχνευσης αντικειμένων είναι τα καλά αποτελέσματα που πέτυχαν συγκεκριμένες αρχιτεκτονικές δικτύων στην εργασία της ανίχνευσης και αναγνώρισης αντικειμένων στον διαγωνισμό IVL SRC [1, 5, 10].



Σχήμα 2.1 – Συνελικτικό Δίκτυο Αναγνώρισης Προσώπου

Learning From Experience

Deep neural networks learn by adjusting the strengths of their connections to better convey input signals through multiple layers to neurons associated with the right general concepts.



When data is fed into a network, each artificial neuron that fires (labeled "1") transmits signals to certain neurons in the next layer, which are likely to fire if multiple signals are received. The process filters out noise and retains only the most relevant features.

Σχήμα 2.2 – Επίπεδα Συνελικτικού Δικτύου και τι Αναγνωρίζει το Καθένα (Εργασία Αναγνώρισης Ζώου)

Κεφάλαιο 3

Μηχανική Όραση και Μηχανική Μάθηση

3.1 Μηχανική Όραση	1
3.2 Τεχνητή Νοημοσύνη	
3.2.1 Γνωστικά και Έμπειρα Συστήματα Μάθησης στην Τεχνητή Νοημοσύνη	
3.3 Μηχανική Μάθηση	
2.3.1 Μάθηση	
3.3.1 Τεχνητά Νευρωνικά Δίκτυα – Ορισμός και Ιστορική Αναδρομή	
3.3.1.1 Τεχνητά Νευρωνικά Δίκτυα – Δίκτυα	
3.3.1.1.1 Βαθιά Συνελεικτικά Δίκτυα - Δίκτυα	
3.3.1.1.2 Αρχιτεκτονικές Βαθιών Συνελεικτικών Δικτύων	
3.3.1.1.3 Βαθιά Συνελεικτικά Δίκτυα - Σημαντικοί Αλγόριθμοι	
3.3.1.1.4 Βαθιά Συνελεικτικά Δίκτυα - Αλγόριθμοι που Χρησιμοποιήθηκαν	
3.3.1.2 Τεχνητά Νευρωνικά Δίκτυα – Σημαντικοί Αλγόριθμοι	
3.3.1.3 Optimizers	
3.6 Αναγνώριση, Ανίχνευση, Κατηγοριοποίηση Αντικειμένων στην Μηχανική Μάθηση και Μηχανική Όραση	

3.1 Μηχανική Όραση

Στην βιβλιογραφία ο όρος computer vision χρησιμοποιείται εναλλακτικά του machine vision. Ένας τυποποιημένος ορισμός που βρέθηκε με την μελέτη της βιβλιογραφίας είναι πως ο σκοπός του computer vision είναι να εξάγει χρήσιμες αποφάσεις για πραγματικά φυσικά αντικείμενα και σκηνές, βάσει των δεδομένων που γίνονται αντιληπτά μέσω μιας εικόνας). Η μηχανική όραση είναι η επιστήμη της όρασης. Η επιστήμη αυτή είναι συνδεδεμένη με την βιολογική όραση. Ο όρος machine vision αναφέρεται στην μελέτη των μεθόδων, των τεχνικών και του υλικού μέσω των οποίων μπορεί να επιτευχθεί η τεχνητή όραση για πρακτικές εφαρμογές [9].

Σύμφωνα με τους ερευνητές, τα τελευταία χρόνια το computer vision αποτελεί τεχνολογία κλειδί, σε πάρα πολλούς τομείς παρόλο που πρόκειται για ένα νέο πεδίο έρευνας. Οι εφαρμογές του εν λόγω πεδίο στην καθημερινότητα μας είναι πάρα πολλές και ανήκουν σε πολλούς και διαφορετικούς τομείς της ζωής μας. Χαρακτηριστικές εφαρμογές αποτελούν η δυνατότητα των σύγχρονων αυτοκινήτων να αναγνωρίσουν αυτόματα την λωρίδα κυκλοφορίας στην οποία βρίσκονται, η αυτόματη αναγνώριση προσώπου μέσω της φωτογραφικής μηχανής του έξυπνου κινητού τηλεφώνου, η αυτόματης αναγνώριση/ανάγνωση λωρίδας γραμμωτού κώδικα (bar code readers), κλπ [9].

Σημαντική είναι όμως και η εκμετάλλευση των δεδομένων που δύναται να εξάγει ένα σύστημα, μέσω της εφαρμογής των τεχνικών του computer vision, τα οποία μπορεί στη συνέχεια να δρομολογηθούν για στρατιωτική χρήση (πχ επιτήρηση ευαίσθητων χώρων), αστυνόμευση δημοσίων χώρων (πχ αναγνώριση συμπεριφοράς πλήθους), αύξηση της απόδοσης της βιομηχανικής παραγωγής (πχ αυτοματοποιημένος έλεγχος) κλπ [9].

Πρώτα οι ερευνητές μελέτησαν έννοιες πως μπορεί να γίνει η τμηματοποίηση της εικόνας και η ανίχνευση των άκρων της. Ακολούθως, την δεκαετία του 1980, η έρευνα γύρω από το πεδίο του computer vision σημείωσε εκθετική πρόοδο λόγω της ανακάλυψης νέων

θεωριών και εννοιών [9].

Στις αρχές της δεκαετίας του 70, οι ερευνητές στα πεδία της τεχνητής νοημοσύνης και της ρομποτικής, στα πανεπιστήμια Stanford, MIT και CMU, προσπαθούσαν να επιλύσουν το πρόβλημα της αυτόματης αναγνώρισης αντικειμένων με τη χρήση εικόνων που κατέγραφε μια κάμερα. Το πεδίο της επεξεργασίας εικόνας εκείνη την περίοδο ήταν σε ακμή και για να διαχωριστεί το πεδίο του computer vision από αυτήν, ο στόχος του Computer Vision διαφοροποιήθηκε ώστε να αποτελεί την προσπάθεια για τρισδιάστατη αναπαραγωγή του χώρου σύμφωνα με πληροφορίες που λαμβάνονταν από εικόνες [9].

Κατά τη δεκαετία του 1990 το πεδίο του computer vision ήταν στενά συνδεδεμένο με το πεδίο του machine vision. Τότε σημειώθηκε πρόοδος στην αρχιτεκτονική ελέγχου και την λειτουργία των αισθητήρων, εντείνοντας τις δυνατότητες των πιο πάνω συστημάτων ενώ ταυτόχρονα μείωναν το κόστος τους. Ακόμη, μέσα σε αυτήν την δεκαετία ανακαλύφθηκαν οι πρώτες στατιστικές μέθοδοι αναγνώρισης προσώπων σε εικόνες. Το υπέρτατο στοιχείο ακμής αυτής της δεκαετίας ήταν η αλληλεπίδραση με τα γραφικά των υπολογιστών [9].

Μια πρώτη εφαρμογή στην μηχανική όραση ήταν η οπτική αναγνώριση χαρακτήρων η οποία χρησιμοποιήθηκε για την αυτόματη αναγνώριση γραμμάτων, συμβόλων και αριθμών.

Μεταξύ των ετών. 2000 και 2010 αναπτύχθηκαν αλγόριθμοι μηχανικής μάθησης και τεχνικές που στηρίζονται στα χαρακτηριστικά ενός αντικειμένου με στόχο την αυτόματη αναγνώριση αντικειμένων [9].

Κατά την δεκαετία που προαναφέραμε ήταν καθοριστική η αύξηση της χρήσης του διαδικτύου που σε συνδυασμό με την αύξηση του εύρους ζώνης οδήγησαν στην ανάπτυξη μεγάλων ταχυτήτων μεταφοράς δεδομένων που είναι σημαντικός παράγοντας για την ανάπτυξη και την χρήση αλγορίθμων μηχανικής όρασης για την αναγνώριση και ανίχνευση αντικειμένων καθώς και για την μετακίνηση βάσεων δεδομένων με εικόνες [9].

Το 2010, αρκετές εικόνες που βρίσκονται σε βάσεις δεδομένων επισημαίνονται και περιέχουν πλέον χρήσιμα στοιχεία όπως η θέση των αντικειμένων και έτσι η διαδικασία εκπαίδευσης βαθιών νευρωνικών δικτύων να γίνεται πιο εύκολη αφού κατά την διάρκειά της δεν απαιτείται συνεχής παρακολούθηση από άνθρωπο [9].

3.2 Τεχνητή Νοημοσύνη

Η τεχνητή νοημοσύνη είναι ένα πεδίο της πληροφορικής που άνθισε στα τέλη της δεκαετίας του 50. Το πεδίο αυτό ασχολείται με την δημιουργία αλγορίθμων και συστημάτων που να μπορούν να σκεφτούν τις λύσεις σε σημαντικά προβλήματα [18].

Ο όρος τεχνητή νοημοσύνη αναφέρεται σε συστήματα και μηχανές που είναι έξυπνα. Τα συστήματα αυτά είναι νοήμονα - μπορούν να σκέφτονται - συλλέγουν και να επεξεργάζονται πληροφορίες με σκοπό την απόκτηση γνώσης [18].

3.2.1 Γνωστικά και Έμπειρα Συστήματα Μάθησης στην Τεχνητή Νοημοσύνη

Η μάθηση είναι ένα κύριο χαρακτηριστικό της νοήμονος συμπεριφοράς ενός ανθρώπου. Στην τεχνητή νοημοσύνη ένα σύστημα μάθησης είναι ένας πράκτορας / μια οντότητα που καλείται να μάθει το περιβάλλον στο οποίο βρίσκεται με τον πιο έξυπνο τρόπο [18].

Η μάθηση στην τεχνητή νοημοσύνη επιτυγχάνεται μέσω της αλληλεπίδρασης του πράκτορα με το περιβάλλον του. Ο πράκτορας σε κάθε βήμα του, συλλέγει δεδομένα που σχετίζονται με το πρόβλημα που θέλει να λύσει. Ο στόχος του πράκτορα είναι βελτιώσει τον τρόπο που εκτελεί μια συγκεκριμένη ενέργεια ως προς μια μετρική με την επανάληψη της σύμφωνα με τις γνώσεις που έχει αποκτήσει. Ως εκ τούτου, δεν χρειάζεται να

προγραμματιστεί εκ νέου για να εκτελέσει μια συναφή ενέργεια διότι μπορεί να αξιοποιήσει την εμπειρία του [18].

3.3 Μηχανική Μάθηση

Η μηχανική μάθηση στην επιστήμη των υπολογιστών είναι ένα τα πεδίο που άκμασε από τη μελέτη της υπολογιστικής θεωρίας μάθησης στην τεχνητή νοημοσύνη και της αναγνώρισης προτύπων. Σύμφωνα με ένα ορισμό η μηχανική μάθηση είναι το πεδίο μελέτης που δίνει στους υπολογιστές της δυνατότητα να μαθαίνουν να αναγνωρίζουν πρότυπα και να εκτελούν συγκεκριμένες εργασίες χωρίς να έχουν προγραμματιστεί ρητά. Η λέξη μηχανική αναφέρεται στο είδος του πράκτορα που λαμβάνει μέρος στη διαδικασία της μάθησης, στην περίπτωση μας ο υπολογιστής-μηχανή. Η λέξη μάθηση αναφέρεται στην διαδικασία που ακολουθεί ο πράκτορας για να μάθει το περιβάλλον. Το περιβάλλον είναι μερικώς γνωστό ή εντελώς άγνωστο. Ένα σύστημα μηχανικής μάθησης είναι ένα σύστημα που βελτιώνει την επίδοσή του σε μια συγκεκριμένη εργασία σύμφωνα με συγκεκριμένη μετρική ή μετρικές. Η μάθηση πραγματοποιείται με την εκπαίδευση του δικτύου σε πρότυπα που σχετίζονται με μια συγκεκριμένη εργασία με στόχο την αναγνώριση νέων προτύπων τα οποία το δίκτυο δεν έχει δει πιο πριν. Ένας παρόμοιος ορισμός ορίζει την μηχανική μάθηση ως την μελέτη των κόμβων/νευρώνων που προσαρμόζονται μέσω μάθησης, αλλάζουν την εμπειρία που έχουν αποθηκευμένη μέσω μάθησης και την χρησιμοποιούν για να κάνουν διάφορες εργασίες [16, 17].

Τεχνητά Νευρωνικά Δίκτυα

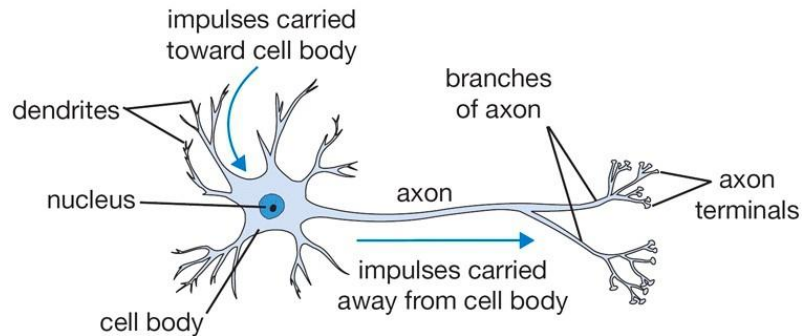
Τα Τεχνητά Νευρωνικά συστήματα είναι υπολογιστικά συστήματα βασισμένα στα βιολογικά νευρωνικά συστήματα. Τα συστήματα αυτά μαθαίνουν συγκεκριμένα μοτίβα και εργασίες. Ένα σύστημα μάθησης μπορεί να περιγράψει ως ένα σύστημα που σταδιακά

βελτιώνει την επίδοση του στην εκτέλεση μιας συγκεκριμένης εργασίας σύμφωνα με συγκεκριμένες μετρικές. Η μάθηση επιτυγχάνεται μέσω της εκπαίδευσης του δικτύου με παραδείγματα και δεδομένα που σχετίζονται με την εργασία που θέλουμε να μάθει. Ο στόχος του δικτύου αυτού είναι να μπορεί να κατηγοριοποιήσει σωστά νέα δεδομένα που δεν είδε πριν. Τα δίκτυα αυτά έχουν την ικανότητα να εξάξουν χρήσιμες πληροφορίες από πολλά δεδομένα και μπορούν να γενικεύσουν καλά χρησιμοποιώντας προηγούμενα παραδείγματα. Διαθέτουν ελάχιστες υπολογιστικές απαιτήσεις όταν είναι πλήρως εκπαιδευμένα λόγω της αρχιτεκτονικής τους. Αυτό οφείλεται στο γεγονός πως μπορούν να χρησιμοποιήσουν τον παραλληλισμό που βελτιώνει την ταχύτητα εκπαίδευσης σε αντιπαραβολή με την εκτέλεση παραδοσιακών σειριακών υπολογισμών. Τέλος, είναι η κατάλληλη επιλογή όταν θέλουμε να επιλύσουμε προβλήματα που δεν είναι πλήρως ορισμένα ή χρειάζονται μεγάλο πλήθος δεδομένων για να επιλυθούν [16].

Αρχικά, τα τεχνητά νευρωνικά συστήματα δημιουργήθηκαν με στόχο να προσομοιώσουν τον ανθρώπινο εγκέφαλο και να λύσουν προβλήματα με τον τρόπο που θα τα έλυνε αυτός. Αυτό φαίνεται στη βασική αρχιτεκτονική ενός τεχνητού νευρωνικού δικτύου. Παρουσιάζουμε πιο κάτω τις αρχιτεκτονικές των τεχνητών και των βιολογικών νευρωνικών δικτύων ώστε να καταλάβουμε τις ομοιότητες τους [16].

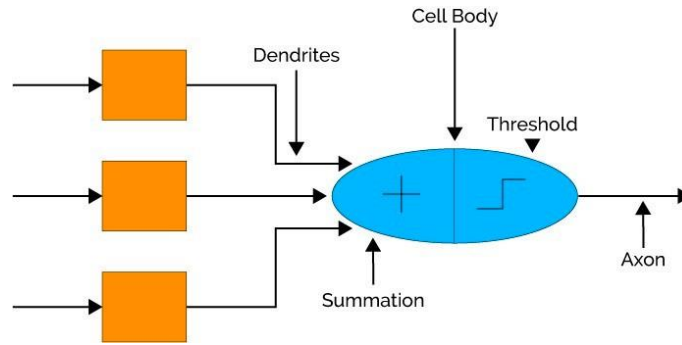
Ένα βιολογικό νευρωνικό δίκτυο είναι μια συλλογή από νευρώνες που λαμβάνουν, επεξεργάζονται και μεταδίδουν πληροφορίες μεταξύ τους μέσω ηλεκτρικών και χημικών σημάτων χρησιμοποιώντας ειδικές συνδέσεις τις συνάψεις. Το Τεχνητό Νευρωνικό Δίκτυο έχει την ίδια αρχιτεκτονική με το βιολογικό νευρωνικό δίκτυο, απαρτίζεται από τεχνητούς νευρώνες – κόμβους, μια απλούστερη έκδοση βιολογικών νευρώνων όσον αφορά την λειτουργικότητα τους, ακμές ή συνδέσεις. Δεν έχει συνάψεις που μεταφέρουν σήματα από ένα τεχνητό νευρώνα στον άλλον. Ένας βιολογικός νευρώνας αποτελείται από τρία συστατικά, το σώμα του κυττάρου, τους άξονες και τους δεντρίτες. Τα σήματα μεταφέρονται διαμέσου των αξόνων και λαμβάνονται από τους δεντρίτες. Ακολούθως, οι δεντρίτες μεταδίδουν το σήμα στο σώμα του κυττάρου. Στη συνέχεια, το σώμα του

κυττάρου που είναι υπεύθυνο για την επεξεργασία των σημάτων για να τα συναθροίσει μαζί. Τέλος, αν η συναθροισμένη τιμή ξεπερνά την τιμή ενός προκαθορισμένου κατωφλίου τότε ο νευρώνας πυροδοτεί ένα σήμα και το στέλνει σε ένα άλλο νευρώνα με τον οποίον είναι συνδεδεμένος. Τα τεχνητά νευρωνικά δίκτυα λειτουργούν με παρόμοιο τρόπο [16].



Σχήμα 3.1 – Μοντέλο Βιολογικού Νευρώνα

Η ιδέα αναπαραγωγής της λειτουργίας του ανθρώπινου εγκεφάλου ήταν αυτή που κίνησε το ενδιαφέρον των ερευνητών να ασχοληθούν με τα τεχνητά νευρωνικά δίκτυα. Όμως, σταδιακά, η ανάγκη επίλυσης συγκεκριμένων εργασιών τους ώθησε στην ανάπτυξη νευρωνικών δικτύων με διαφορετικές αρχιτεκτονικές. Μερικές από αυτές τις αρχιτεκτονικές δικτύων δεν θυμίζουν καθόλου το βιολογικό νευρωνικό δίκτυο που αποτέλεσε την πηγή έμπνευσης των νευρωνικών δικτύων. Ένα παράδειγμα αυτής της νέας αρχιτεκτονικής, είναι η ανάπτυξη μιας εναλλακτικής αρχιτεκτονικής τεχνητού νευρωνικού δικτύου, το βαθύ συνελκτικό δίκτυο που ενσωματώνει μια μονάδα προεπεξεργασίας που εξάγει πολύπλοκα όμως χρήσιμα χαρακτηριστικά από εικόνες, βίντεο, ήχο. Στη συνέχεια, αυτά τα χαρακτηριστικά δίνονται σε ένα παραδοσιακό τεχνητό νευρωνικό δίκτυο, ένα πλήρως συνδεδεμένο δίκτυο εμπρόσθιου περάσματος MLP που κατηγοριοποιεί και βάζει μια ετικέτα στην αρχική εικόνα. Υπάρχουν πάρα πολλά είδη και παραλλαγές των νευρωνικών δικτύων για την επίλυση συγκεκριμένων εργασιών όπως αναγνώριση ομιλίας, αναγνώριση αντικειμένων, αυτόματη μετάφραση κειμένου από τεχνητά νευρωνικά δίκτυα κλπ. Θα παρουσιάσουμε μερικά από τα είδη των δικτύων στις επόμενες παραγράφους [16,17].



Σχήμα 3.2 – Αναλογία τεχνητού Νευρώνα με βιολογικό νευρώνα.

3.3.1 Μάθηση

Η μάθηση στα νευρωνικά συστήματα μάθησης βρίσκεται αποθηκευμένη στα βάρη ή τις συνδέσεις μεταξύ των κόμβων/νευρώνων. Τα βάρη αλλάζουν μέσω της εμπειρίας (μάθησης) που αποκτά το νευρωνικό δίκτυο, σύμφωνα με ένα κανόνα ενημέρωσης για μια αλλαγή ενός βάρους.

3.3.2 Κλάσεις Μάθησης:

3.3.2.1 Επιβλεπόμενη Μάθηση

Ένας δάσκαλος δίνει τα δεδομένα εισόδου και λέει στο δίκτυο ποια είναι τα αντίστοιχα επιθυμητά αποτελέσματα για το κάθε δεδομένο εισόδου. Τα βάρη που υπάρχουν στο δίκτυο ενημερώνονται έτσι ώστε να ελαχιστοποιούν τη διαφορά μεταξύ του επιθυμητού και του πραγματικού αποτελέσματος για κάθε δεδομένο εισόδου [12, 16].

3.3.2.2 Ενισχυτική Μάθηση

Ένας κριτής βλέπει την πράξη που έκανε το νευρωνικό δίκτυο και επιβραβεύει το δίκτυο για την πράξη του αν αυτή φέρνει το δίκτυο πιο κοντά στον στόχο του ή το τιμωρεί για αυτή αν αυτή το απομακρύνει από τον στόχο του. Τα βάζα του δικτύου ενημερώνονται με τέτοιο τρόπο ώστε να αναπτυχθεί μια σχέση μεταξύ των δεδομένων εισόδου/εξόδου που να μεγιστοποιεί την πιθανότητα να λάβει επιβράβευση και να ελαχιστοποιήσει την πιθανότητα να λάβει τιμωρία στο επόμενο βήμα [12, 16].

3.3.2.3 Μη Επιβλεπόμενη Μάθηση

Το δίκτυο μπορεί να ομαδοποιήσει στοιχεία σύμφωνα με κοινά χαρακτηριστικά που έχουν τα δεδομένα εισόδου. Συγκεκριμένα, το νευρωνικό δίκτυο αναπτύσσει διαφορετικές συμπεριφορές για να αναπαραστήσει τις διαφορετικές κλάσεις που βρίσκονται στα δεδομένα εισόδου (στις πρακτικές εφαρμογές είναι απαραίτητη η χρήση ετικετών μετά την εκπαίδευση ώστε να μπορεί να αντιστοιχίσει την ορθή κλάση της εφαρμογής μας με την ορθή συμπεριφορά) [12, 16].

3.3.2 Τεχνητά Νευρωνικά Δίκτυα - Ορισμός και Ιστορική Αναδρομή

Στις αρχές της δεκαετίας του 1940, οι McCulloch & Pitts πρότειναν ένα μοντέλο που έκανε απλές πράξεις. Στα τέλη εκείνης της δεκαετίας, ο Hebb πρότεινε την μάθηση με συσχετίσεις γνωστή και ως Hebbian cell assemblies [16].

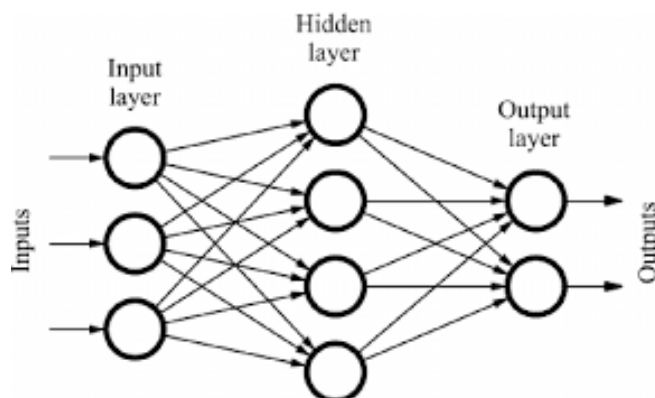
Αργότερα, στις αρχές της δεκαετίας του 1950 ανακαλύφθηκε ο Hebbian κανόνας μάθησης σύμφωνα με τον οποίο όταν δύο νευρώνες είναι ενεργοί ταυτόχρονα τότε η σύνδεσή τους γίνεται πιο δυνατή. Το 1951, ο Minsky εκδίδει την έρευνά του στην ενισχυτική μάθηση και έγινε ο πρώτος ερευνητής που μίλησε για την ενισχυτική μάθηση [16].

Την επόμενη δεκαετία ο Rosenblatt περιγράφει την διαδικασία μάθησης Perceptron (προσαρμογή της σύνδεσης σε ένα νευρώνα/κόμβο επαναληπτικά όταν η απάντηση του νευρώνα/κόμβου είναι λανθασμένη. Ο Rosenblatt και οι υπόλοιποι ερευνητές εκείνης της δεκαετίας πίστευαν ότι το Perceptron μπορούσε να λύσει κάθε είδους πρόβλημα [16].

Στα τέλη της δεκαετίας του 1960, ο Minsky και ο Papert εκδίδουν ένα βιβλίο με το οποίο αποδεικνύουν πως τα Perceptron δεν είναι ικανά να λύσουν οποιοδήποτε πρόβλημα. Αντίθετα, τα δίκτυα Perceptrons μπορούσαν να επιλύσουν μόνο ένα περιορισμένο αριθμό προβλημάτων. Παραδείγματος χάρη τα δίκτυα αυτά δεν μπορούσαν να λύσουν γραμμικά και μη διαχωρίσιμα προβλήματα. Ακόμη, ο Minsky και ο Papert προέβλεψαν ότι τα Perceptrons δεν θα μπορούσαν να επεκταθούν ακόμα και αν χρησιμοποιούνταν αρχιτεκτονικές πολλαπλών επιπέδων μάθησης. Ως εκ τούτου, η χρηματοδότηση για την έρευνα στο πεδίο της μηχανικής μάθησης παύει να χορηγείται [16].

Στα μέσα της δεκαετίας του 1970, οι Werbos ανακαλύπτουν τον αλγόριθμο μάθησης ανάστροφης μετάδοσης σφάλματος. Τα πολλαπλών επιπέδων εμπρόσθιου περάσματος δίκτυα δεν έγιναν γνωστά στο ευρύ κοινό εκείνη την εποχή [16].

Μετά από δέκα χρόνια, στα μέσα περίπου της δεκαετίας του 1980, ο Rumelhart και άλλοι ερευνητές ανακάλυψαν ξανά τον αλγόριθμο ανάστροφης μετάδοσης σφάλματος και άλλες διαδικασίες μάθησης για πολυστρωματικά νευρωνικά δίκτυα. Ακόμη, το 1982, ανακαλύφθηκε το δίκτυο Hopfield. Το δίκτυο αυτό χρησιμοποιεί μια ενεργειακή επιφάνεια που αποθηκεύει ενέργεια στα τοπικά ελάχιστα. Την ίδια χρονιά έγιναν γνωστά τα δίκτυα Kohonen (αυτά είχαν πρωτοανακαλυφθεί στις αρχές της δεκαετίας του 1970). Αυτός ο αλγόριθμος βασίζεται στην ιδέα ότι οι νευρώνες μπορούν να αυτό-οργανωθούν από μόνοι τους ώστε να (tune) διάφορα και συγκεκριμένα μοτίβα [16].



Σχήμα 3.3 Νευρωνικό Δίκτυο με ένα Κρυφό Επίπεδο Νευρώνων. Δέχεται τρεις Κατηγορίες Δεδομένων Εισόδου και τις χωρίζει σε δύο Κατηγορίες στην Έξοδο.

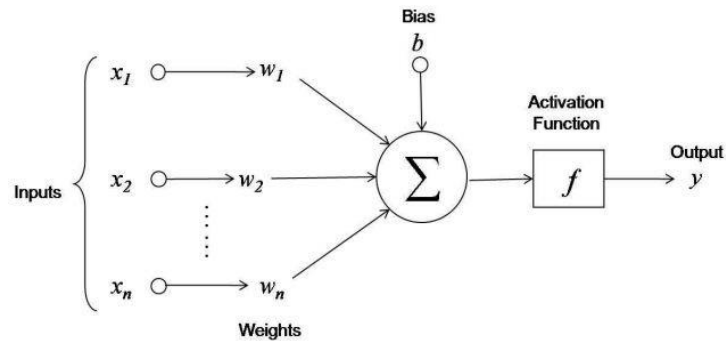
Τεχνητά Νευρωνικά Δίκτυα – Δίκτυα

Σε αυτή την υποενότητα, θα παρουσιάσουμε και θα εξηγήσουμε τις κλασικές αρχιτεκτονικές νευρωνικών δικτύων.

ΤΝΔ McCulloch and Pitts (McP)

Το δίκτυο McP αποτελεί τον θεμέλιο λίθο των Τεχνητών Νευρωνικών Δικτύων. Δημιουργήθηκε από τους Warren McCulloch και Walter Pitts το 1943 και ονομάστηκε ως Binary Threshold Unit. Το μοντέλο αυτό, είχε ως στόχο την αναπαραγωγή την δομής και της λειτουργίας ενός βιολογικού νευρώνα του νευρωνικού δικτύου που βρίσκεται στον ανθρώπινο εγκέφαλο. Με βιολογικούς όρους, ένας πίνακας εισόδου μπαίνει στην θέση των δεντρίτων και στέλνει στο ΤΝΔ σήματα, εκτελώντας πολλαπλασιασμούς με τις τιμές των βαρών που βρίσκονται στις ακμές του δικτύου. Ο τεχνητός νευρώνας αθροίζει τα σήματα και μεταδίδει την συναθροισμένη τιμή στην συνάρτηση ενεργοποίησης.- Συνάρτηση Σκαλί). Στην συνάρτηση Σκαλί αν, η συναθροισμένη τιμή ξεπεράσει μια συγκεκριμένη οριοθετημένη τιμή, το σήμα εξόδου της γίνεται 1 (πυροδοτεί) αλλιώς το

σήμα εξόδου της γίνεται 0 (δεν πυροδοτεί)(όπως ακριβώς γίνεται και στους βιολογικούς νευρώνες). Επομένως, μπορεί να χρησιμοποιηθεί για δυαδική κατηγοριοποίηση [16].



Σχήμα 3.4 - Αναπαράσταση Τεχνητού Νευρώνα McP

$$y = \begin{cases} 1 & , if wx > s \\ 0 & , otherwise \end{cases}$$

Εξίσωση 3.1 - Συνάρτηση (Ενεργοποίησης) Σκαλί – όπου y η έξοδος του δικτύου, x το διάνυσμα εισόδου, w το διάνυσμα βάρους βαρών, wx το εσωτερικό γινόμενο και s το κατώφλι.

Ο τρόπος με τον οποίο κατηγοριοποιείται η είσοδος εξαρτάται από τα βάρη που βρίσκονται πάνω στις συνδέσεις του δικτύου και από την οριοθετημένη τιμή της συνάρτησης ενεργοποίησης. Σε ένα από διδιάστατο χώρο (δύο διαστάσεων πίνακα εισόδου) η γραμμή απόφαση είναι η εξίσωση 3.2 [16].

$$x_2 = -(w_1 w_2) x_1 + s w_2$$

Εξίσωση 3.2: Γραμμή Απόφασης ενός Διδιάστατου Πίνακα Εισόδου

T.N.Δ Perceptron

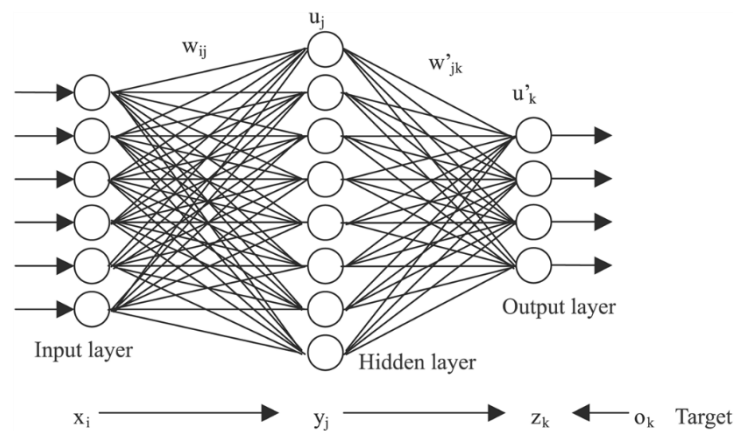
Το Perceptron είναι ένα τεχνητό νευρωνικό δίκτυο που ανακαλύφθηκε από τον Rosenblatt το 1957. Είναι δίκτυα εμπρόσθιου περάσματος που απαρτίζονται από τουλάχιστον δύο επίπεδα, επίπεδο νευρώνων εισόδου, επίπεδο νευρώνων εξόδου και ένα μόνο κρυφό επίπεδο νευρώνων. Οι νευρώνες που βρίσκονται στο κρυφό επίπεδο είναι νευρώνες McP. Το κρυφό επίπεδο σε ένα Perceptron δεν είναι απαραίτητο επίπεδο και μπορεί να

παραληφθεί [16].

Το δίκτυο αποτελεί την απλούστερη discriminant συνάρτηση με εσωτερικές παραμέτρους που μπορεί να εκπαιδευτεί. Σε αυτήν η είσοδος συνδυάζεται γραμμικά και στην συνέχεια τροφοδοτείται σε μια μη γραμμική συνάρτηση όπως την $\gamma = f(\sum_i a_i x_i)$. Το Perceptron μπορεί να επιλύσει μόνο γραμμικά διαχωρίσιμα προβλήματα (τα προβλήματα που οι κλάσεις τους είναι γραμμικά διαχωρίσιμες). Με την χρήση ενός μόνο νευρώνα Perceptron μπορούμε να λύσουμε ένα γραμμικά διαχωρίσιμο πρόβλημα. Ακόμη, δεν μπορεί να κατηγοριοποιήσει δεδομένα που έχουν variance π.χ. μια μετατόπιση της εικόνας εισόδου [16].

Δίκτυα με ένα κρυφό Επίπεδο και Γραμμικά Διαχωρίσιμο Πρόβλημα

Δύο κλάσεις που δεν είναι γραμμικά διαχωρίσιμες στον αρχικό χώρο αναπαράστασης δεδομένων τους γίνονται γραμμικά στον χώρο αναπαράστασης δεδομένων του κρυφού επιπέδου λόγω ενός μη γραμμικού μετασχηματισμού που εφαρμόζεται σε αυτά [16].



Σχήμα 3.5 - Νευρωνικό Δίκτυο με ένα Κρυφό Επίπεδο Νευρώνων

Δίκτυο Πολυστρωματικών Επίπεδων Perceptron

Το νευρωνικό δίκτυο πολυστρωματικών επιπέδων Perceptron είναι ένα εμπρόσθιο δίκτυα Perceptron που περιέχει πολλά επίπεδα από πολλούς McP νευρώνες. Με τον όρο εμπρόσθιο δίκτυο εννοούμε πως η πληροφορία μεταφέρεται προς τα μπροστά (από το επίπεδο εισόδου στα κρυφά επίπεδα και τέλος στο επίπεδο εξόδου). Οι νευρώνες McP χρησιμοποιούν μια συνάρτηση κατωφλίου ως συνάρτηση ενεργοποίησης (βηματική συνάρτηση ή συνάρτηση σκαλί) ενώ οι Νευρώνες πολυστρωματικών επιπέδων Perceptron χρησιμοποιούν μια αυθαίρετη συνάρτηση ενεργοποίησης. Επομένως, τα δίκτυα McP επιλύουν προβλήματα δυαδικής κατηγοριοποίησης ενώ τα δίκτυα MLP επιλύουν είτε προβλήματα πρόβλεψης είτε προβλήματα κατηγοριοποίησης σύμφωνα με την συνάρτηση ενεργοποίησης που χρησιμοποιούν. Μπορούμε να πούμε πως οι συναρτήσεις ενεργοποίησης είναι ένας μηχανισμός με τον οποίο μπορούμε να ξεχωρίσουμε τις εξόδους που βρίσκονται πιο κοντά στην επιθυμητή κλάση, και μας δίνει μια αίσθηση του πόσο πρέπει να προσαρμόσουμε τα βάρη ώστε το δίκτυο μας να μας δίνει καλύτερες προβλέψεις ή κατηγοριοποιήσεις. Τα δίκτυα αυτά όπως και τα δίκτυα Perceptron με ένα κρυφό επίπεδο έχουν ένα επίπεδο εισόδου, ένα επίπεδο εξόδου και τουλάχιστον ένα κρυφό επίπεδο που διαθέτει ένα ή περισσότερους κρυφούς νευρώνες. Κάθε επίπεδο εκτός του επιπέδου εξόδου έχει μια ανεξάρτητη μονάδα νευρώνα το bias. Οι νευρώνες του κρυφού επιπέδου είναι οι νευρωνικές μονάδες που αποθηκεύουν και ενημερώνουν της πληροφορίες του δικτύου κατά την εκπαίδευση του δικτύου. Κάθε νευρωνική μονάδα στο κρυφό επίπεδο ορίζει μια νέα γραμμή απόφασης που διαχωρίζει τα δεδομένα σε κλάσεις. Επίσης, η προσθήκη δεύτερου κρυφού επιπέδου οδηγεί στον σχηματισμό αυθαίρετων πολύπλοκων επιφανειών απόφασης που μπορούν να διαχωρίσουν οποιεσδήποτε κλάσεις. Η πολυπλοκότητα των σχημάτων αυτών περιορίζεται από το πλήθος των νευρώνων που υπάρχουν στο δίκτυο. Ως εκ τούτου, δεν χρειάζονται περισσότερα από δύο κρυφά επίπεδα (Θεώρημα Kolmogorov). Η διαδικασία υπολογισμού του σήματος εξόδου στο MLP δίκτυο είναι παρόμοια με την διαδικασία που ακολουθήσαμε στο McP δίκτυο. Οι νευρώνες εισόδου παίρνουν τις τιμές τους στο πρώτο κρυφό επίπεδο, όπου σύμφωνα με την συνάρτηση ενεργοποίησης που έχουν εκπέμπουν ως έξοδο ένα σήμα. Στην συνέχεια, αυτά

τα σήματα δίνονται ως είσοδο στο επόμενο κρυφό επίπεδο ή επίπεδο εξόδου όπου η διαδικασία επαναλαμβάνεται μέχρι να φτάσουμε στο τελευταίο επίπεδο του δικτύου:

$$y = f(\sum(wrx + b))$$

Εξίσωση 3.3 – Έξοδος McP Δικτύου

Η έξοδος y ενός νευρώνα όπου το x είναι το διάνυσμα εισόδου για αυτό τον νευρώνα, το w είναι το διάνυσμα των βαρών, το wrx είναι το εσωτερικό γινόμενο, b το κατώφλι και s μια αυθαίρετη συνάρτηση ενεργοποίησης [16].

Αλγόριθμος Κατάβαση Κλίσης

Η Κατάβαση Κλίσης είναι ένας μαθηματικός αλγόριθμος βελτιστοποίησης για την εύρεση μιας ελάχιστης συνάρτησης. Είναι ένας από τους πιο πολυχρησιμοποιημένους αλγόριθμους στο πεδίο των Τεχνητών Νευρωνικών Δικτύων. Ο αλγόριθμος αυτός χρησιμοποιείται για την εκπαίδευση νευρωνικών δικτύων. Ο τρόπος με τον οποίο δουλεύει περιγράφεται από το όνομά του: δοθέντος ενός σημείου, κάνε βήματα ανάλογα του αρνητικού της κλίσης της συνάρτησης (στο σημείο που βρίσκεται). Ένα σήμα σφάλματος είναι το σήμα που περιγράφει πόσο καλά ή κακά το δίκτυο μας κατηγοριοποιεί τα δεδομένα εισόδου του. Ο στόχος μας είναι να ελαχιστοποιήσουμε αυτή της συνάρτηση ώστε να ταξινομήσουμε σωστά όσα περισσότερα δεδομένα εισόδου μπορούμε [16].

$$E = \frac{1}{2} \sum (Tp_j - Op_j)^2$$

Εξίσωση 3.4 Συνάρτηση Μέσου Τετραγωνικό Σφάλματος όπου το t είναι το επιθυμητό αποτέλεσμα, το o το πραγματικό αποτέλεσμα, το p το μοτίβο και το j ο νευρώνας [16].

Η ιδέα είναι η εφαρμογή του αλγορίθμου μάθησης κατάβασης κλίσης στην συνάρτηση Μέσου τετραγωνικού Σφάλματος ώστε να ελαχιστοποιήσουμε το σφάλμα, δηλαδή την διαφορά μεταξύ του επιθυμητού και του πραγματικού αποτελέσματος [16].

Αυτό σημαίνει πως όταν γίνει μια μεταβολή στο διάνυσμα των βαρών είναι ανάλογη του αρνητικού της παραγώγου του σφάλματος στο τρέχον μοτίβο σε αντιστοιχία με κάθε βάρος [16]:

$$\Delta w_{ij} = -\eta \Delta p \Delta w_{ij}$$

Εξίσωση 3.5 Η μεταβολή των βαρών σύμφωνα με τον αλγόριθμο κατάβασης κλίσης όπου το η είναι ο ρυθμός μάθησης.

3.2.3 Τεχνητά Νευρωνικά Δίκτυα – Σημαντικοί Αλγόριθμοι

Αλγόριθμος Μηχανικής Μάθησης Perceptron

Οι νευρώνες McP μαθαίνουν μέσω του αλγορίθμου μάθησης Perceptron. Ο αλγόριθμος αυτός κατασκευάστηκε το 1957 από τον Rosenblatt. Η βασική ιδέα αυτού του αλγορίθμου είναι να παρουσιάσουμε στο δίκτυο τα δεδομένα εισόδου και τα επιθυμητά αποτελέσματα για κάθε ένα από αυτά. Στη συνέχεια, υπολογίζουμε το αποτέλεσμα για το συγκεκριμένο δεδομένο εισόδου. Στην περίπτωση λανθασμένης κατηγοριοποίησης (το αποτέλεσμα είναι 0 αλλά θα έπρεπε να είναι 1 και αντίστροφα), προσαρμόζουμε τα βάρη ανάλογα. Αρχικά, οι ερευνητές πίστευαν πως ο αλγόριθμος Perceptron μπορεί να λύσει οποιοδήποτε πρόβλημα αλλά αποδείχθηκε πως μπορεί να λύσει μόνο γραμμικά διαχωρίσιμα μοτίβα. Ως εκ τούτου, τα μοτίβα αυτά ανήκουν σε γραμμικά διαχωρίσιμες κλάσεις. Επομένως, ο αλγόριθμος Perceptron μπορεί να λύσει γραμμικά διαχωρίσιμα προβλήματα. Στα γραμμικά διαχωρίσιμα προβλήματα μια γραμμή ή επιφάνεια απόφασης μπορεί να χωρίσει τα δεδομένα στον δυσδιάστατο ή τον τρισδιάστατο χώρο σε ξεχωριστές κλάσεις. (Μπορούμε να φανταστούμε τον δυσδιάστατο χώρο ως ένα καρτεσιανό σύστημα αξόνων και τον

τρισεδιάστατο χώρο ως ένα δωμάτιο ή το εσωτερικό ενός κουτιού με μήκος, πλάτος και βάθος. [16]

Τα προβλήματα, όπως αυτό της πύλης OR, μπορούν να επιλυθούν με την χρήση του αλγορίθμου Perceptron αφού μια ευθεία γραμμή μπορεί να διαχωρίσει τα αποτελέσματα σε δύο κλάσεις. Ο αλγόριθμος Perceptron δεν μπορεί να επιλύσει προβλήματα στα οποία χρειαζόμαστε περισσότερες ευθείες γραμμές για να χωρίσουμε τα αποτελέσματα σε κλάσεις [16].

Perceptron Learning Algorithm

1) Initialize weights and threshold randomly.

2) Present input and desired output.

3) Calculate actual output.

4) Adapt weights:

- **If output is 0, should be 1:** $w_i(t+1) = w_i(t) + \eta \cdot x_i(t)$
- **If output is 1, should be 0:** $w_i(t+1) = w_i(t) - \eta \cdot x_i(t)$
- **If output is correct :** $w_i(t+1) = w_i(t)$

όπου $0 \leq \eta \leq 1$, ο ρυθμός μάθησης.

Αλγόριθμος Μάθησης Ανάστροφης Μετάδοσης Σφάλματος

Ο αλγόριθμος ανάστροφης μετάδοσης σφάλματος, μεταφέρει το σφάλμα από ένα επίπεδο στο προηγούμενο ξεκινώντας από το τελευταίο επίπεδο. Για να εφαρμοστεί, χρειάζεται πρώτα να γίνει ένα εμπρόσθιο πέρασμα κατά το οποίο θα υπολογιστεί το σφάλμα. Στη συνέχεια, στο πέρασμα προς τα πίσω, το σφάλμα μεταφέρεται από το ένα επίπεδο του δικτύου στο προηγούμενο μεταβάλλοντας τις τιμές των βαρών ανάλογα. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να περάσουν όλες οι τιμές εισόδου μέσα στο δίκτυο αρκετές φορές ώστε να ελαχιστοποιήσει το σφάλμα σε σημείο που να είναι αρκετά μικρό

ή όταν το δίκτυο εκπαιδεύτηκε για πολλές εποχές (εποχή = ένα πέρασμα όλων των πρότυπων εισόδου στο δίκτυο) [16].

Η χρήση του αλγόριθμου κατάβασης κλήσης είναι σημαντική διότι προσαρμόζει τα βάρη του τελευταίου κρυφού επιπέδου προς το επίπεδο εξόδου . Αυτό συμβαίνει επειδή το επιθυμητό αποτέλεσμα πρέπει να είναι γνωστό ώστε να υπολογιστεί το σφάλμα που δεν είναι γνωστό στα κρυφά επίπεδα [16].

Ο αλγόριθμος Ανάστροφης Μετάδοσης Σφάλματος όπου d_{ij} είναι το σήμα σφάλματος του νευρώνα i στο επίπεδο j , y_{ij} είναι το πραγματικό αποτέλεσμα του νευρώνα i στο επίπεδο j και το d_{ik} είναι το d_{ij} αλλά στην προηγούμενη επανάληψη του αλγορίθμου [16].

3.3.1.1.1 Βαθιά Συνελικτικά Δίκτυα και Βαθιά Μάθηση - Ορισμός και Ιστορική Αναδρομή

Βαθύ Συνελικτικό Δίκτυο

Ένα βαθύ συνελικτικό δίκτυο είναι ένα δίκτυο εμπνευσμένο από την οργάνωση του οπτικού εγκεφαλικού φλοιού των ζώων. Στον εγκεφαλικό φλοιό των ζώων, συγκεκριμένα της μύγας, μεμονωμένοι νευρώνες του ματιού ανταποκρίνονται σε ερεθίσματα σε μια περιορισμένη περιοχή του χώρου (receptive field). Τα receptive fields των νευρώνων επικαλύπτονται δημιουργώντας το οπτικό πεδίο. Η ανταπόκριση ενός μεμονωμένου νευρώνα σε ερεθίσματα μέσα στο receptive field μπορεί να προσεγγιστεί μαθηματικά με μια λειτουργία συνέλιξης. Τα βαθιά συνελικτικά δίκτυα χρησιμοποιούνται σε εφαρμογές όπου τα δεδομένα εισόδου είναι σε μορφή εικόνας, ήχου ή βίντεο. Τα δίκτυα αυτά παίρνουν ως είσοδο χαρακτηριστικά των δεδομένων εισόδου και εκτελούν μια συγκεκριμένη εργασία σε αυτά (κατηγοριοποίηση ή πρόβλεψη) π.χ. αναγνώριση ενός συγκεκριμένου αντικειμένου σε μια εικόνα ή ένα βίντεο, αναγνώριση ενός ηχητικού

μοτίβου σε ένα τραγούδι, παραγωγή πρότασης που να περιγράφει μια εικόνα, ανακατασκευή σκίτσου σε πίνακα ζωγραφικής κ.λπ [1, 16].

ImageNet

Το ImageNet είναι έργο που σχεδιάστηκε για το πεδίο έρευνας της οπτικής αναγνώρισης αντικειμένων με την χρήση λογισμικού. Στόχος του έργου αυτού είναι η ετικετοποίηση και η κατηγοριοποίηση εικόνων σε 22 χιλιάδες κατηγορίες σύμφωνα με ένα προκαθορισμένο σύνολο λέξεων ή φράσεων. Το έργο αυτό αποτελείται από 14 εκατομμύρια εικόνες και τις οργανώνει με την χρήση της ιεραρχίας WordNet. Στην ιεραρχία WordNet κάθε χρήσιμη λέξη ή φράση ονομάζεται σύνολο συνωνύμων. Οι εικόνες στο ImageNet οργανώνονται σύμφωνα με το σύνολο συνωνύμων και σκοπός του είναι να συγκεντρώσει περισσότερες από 1000 εικόνες ανά σύνολο συνωνύμων [12].

Το ImageNet διοργανώνει ένα διαγωνισμό λογισμικού, τον ImageNet Large Scale Visual Recognition Challenge όπου προγράμματα λογισμικού διαγωνίζονται για να κατηγοριοποιήσουν και να αναγνωρίσουν σωστά διάφορα αντικείμενα. Ο κύριος στόχος του διαγωνισμού είναι η εκπαίδευση ενός μοντέλου που να είναι σε θέση να κατηγοριοποιήσει πολλές εικόνες, κάθε εικόνα σε ένα από τα 1000 σύνολα συνωνύμων του ImageNet. Οι κανόνες του διαγωνισμού ορίζουν πως το δίκτυο χρειάζεται να εκπαιδευτεί με 1 εκατομμύριο εικόνες, να επαληθευτεί με 50 χιλιάδες και να ελεγχθεί με 100 χιλιάδες εικόνες. Οι κλάσεις κατηγοριοποίησης των εικόνων αναπαριστούν κλάσεις αντικειμένων που συναντούμε καθημερινά όπως οικιακά αντικείμενα, ζώα, οχήματα, λουλούδια κλπ. Οι αρχιτεκτονικές συνελκτικών δικτύων κατέκτησαν τον βαθμολογικό πίνακα του διαγωνισμού από το 2012. Στην παρούσα διπλωματική εργασία, θα χρησιμοποιήσουμε μερικές από αυτές τις αρχιτεκτονικές (VGG16, ResNet50 στην υλοποίηση των εφαρμογών αυτής της διπλωματικής εργασίας [12].

Βαθιά Μάθηση

Βαθιά Μάθηση είναι μια τεχνική που μας βοηθά να εκπαιδεύσουμε συστήματα μηχανικής μάθησης. Με την χρήση αυτής της τεχνικής μπορούμε να αναγνωρίσουμε αντικείμενα μέσα σε εικόνες ή βίντεο, συγκεκριμένους ήχους σε ένα ηχητικό σήμα [1].

Μια ενδιαφέρουσα εφαρμογή της βαθιάς μηχανικής μάθησης σχετικά με δεδομένα εισόδου ηχητικά σήματα είναι η μίξη διαφορετικών υπάρχοντων τραγουδιών με σκοπό την δημιουργία ενός καινούργιου [1].

Η εφαρμογή που έχουμε αναφέρει έχει υλοποιηθεί από ερευνητές της Oracle με στόχο την δημιουργία ενός καινούργιου τραγουδιού για τον Ευρωπαϊκό Διαγωνισμό Μουσικής Eurovision. Συγκεκριμένα, δώθηκαν χιλιάδες παλιά τραγούδια Eurovision σε βαθιά συνελκτικά δίκτυα που με την χρήση ενός αλγορίθμου βαθιάς μάθησης δημιούργησαν καινούργια μουσική και στίχους, συνθέτοντας έτσι το πρώτο «τεχνητό» τραγούδι Eurovision [19].

Ιστορική Αναδρομή

Το πρώτο γνωστό βαθύ συνελκτικικό δίκτυο ήταν το LeNet, ανακαλύφθηκε την δεκαετία του 90 και πήρε το όνομα του από τον Πατέρα των βαθιών συνελκτικών δικτύων και εφευρέτη του Yann LeCun. Η αρχιτεκτονική αυτή περιείχε μόνο ένα επίπεδο συνέλιξης που ακολουθήτουν από ένα επίπεδο POOLING . Το συνελκτικό δίκτυο χρησιμοποιήτουν για αναγνώριση χαρακτήρων όπως ταχυδρομικούς κώδικες, ψηφία κλπ [1, 4, 10].

Την δεκαετία του 2000, τα βαθιά συνελκτικά δίκτυα δεν ήταν δημοφιλής τομέας έτσι δεν αναπτύχθηκε κάποια επώνυμη αρχιτεκτονική εκείνη την περίοδο[16].

Αργότερα το 2012, εμφανίστηκε μια νέα αρχιτεκτονική βαθιού συνελικτικού δικτύου. Ονομάστηκε AlexNet και κατατρόπωσε τους αντιπάλους της στον διαγωνισμό ILSVRC την ίδια χρονιά (top 5 σφάλμα 16% σε αντίθεση με την δεύτερη αρχιτεκτονική δικτύου που έβρισκε 26%). Η αρχιτεκτονική του ήταν παρόμοια με αυτή του LeNet αλλά είχε περισσότερα, βαθύτερα επίπεδα συνέλιξης τοποθετημένα το ένα πάνω στο άλλο [4, 10].

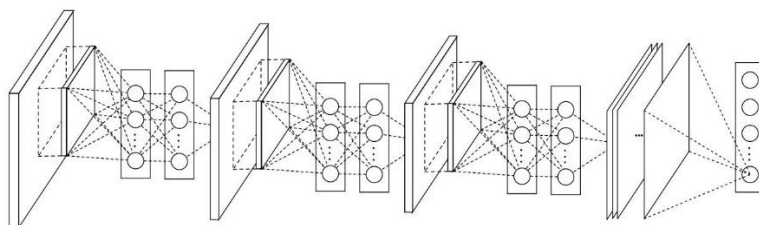
Ένα χρόνο μετά, το 2013, το ZF Net νίκησε τον διαγωνισμό ILSVRC. Η αρχιτεκτονική του δικτύου αυτού βασιζόταν σε αυτήν του AlexNet. Συγκεκριμένα, άλλαξαν το μέγεθος των μεσαίων επιπέδων συνέλιξης κάνοντας μικρότερο το stride και το μέγεθος του φίλτρου στο πρώτο επίπεδο [4, 10].

Το 2014, το VGGNet και το GoogLeNet νίκησαν τον διαγωνισμό ILSVRC. Το VGGNet περιέχει 16 ως 19 επίπεδα συνελίξεων/πλήρως συνδεδεμένα και εφαρμόζει συνελίξεις (convolutions) μεγέθους 3x3 και συγκεντρώσεις (pooling) μεγέθους 2x2. Το VGGNet έχει πάρα πολλές παραμέτρους (140 Million) [1, 2, 4, 10]

Το δίκτυο GoogLeNet ανακαλύφθηκε από την Google και η αρχιτεκτονική του βασίζεται στο βαθύ συνελικτικό δίκτυο LeNet. Το GoogLeNet διαθέτει 22 επίπεδα συνέλιξης. Χρησιμοποιεί μέσο όρο στα επίπεδα συγκέντρωσης του. Η κύρια συνεισφορά του GoogLeNet ήταν η ανάπτυξη του Inception module που μείωσε το πλήθος των παραμέτρων (από 60M στο AlexNet σε 4M στο GoogLeNet. Επομένως, το δίκτυο αυτό εξοικονομεί μνήμη αφού αποθηκεύει λιγότερες παραμέτρους σε σχέση με τα υπόλοιπα δίκτυα. Στη συνέχεια, κατασκευάστηκαν πολλά δίκτυα βασισμένα στο ResNet. Ο νεαρότερος απόγονος του ResNet είναι το δίκτυο Inception-v4 [1, 4, 10].

Το 2015, το βαθύ συνελικτικό δίκτυο ResNet ήταν ο νικητής του διαγωνισμού ILSVRC. Το δίκτυο αυτό δεν περιέχει συγκεκριμένες συνδέσεις και χρησιμοποιεί πάρα πολύ το batch normalization. Επιπρόσθετα, η αρχιτεκτονική αυτή δεν διαθέτει πλήρως συνδεδεμένα

επίπεδα στα τελευταία της επίπεδά ενώ αποτελείται από 152 επίπεδα συνέλιξης. Το ResNet είναι πολύ δημοφιλές μέχρι σήμερα και είναι η πρώτη επιλογή στις πρακτικές εφαρμογές βαθιών συνελικτικών δικτύων [1, 4].



Σχήμα 3.6– Αφαιρετική Αναπαράσταση Αρχιτεκτονικής Συνελικτικών Δικτύων

3.3.1.1.3 Αλγόριθμοι και Optimizers Βαθιών Συνελικτικών Δικτύων που Χρησιμοποιήθηκαν

Σε αυτή την υποενότητα θα παρουσιάσουμε τους αλγόριθμους βελτιστοποίησης που χρησιμοποιήθηκαν κατά την υλοποίηση των εφαρμογών της παρούσας διπλωματικής εργασίας. Στο τρέχον υποκεφάλαιο, θα συζητήσουμε τον αλγόριθμο της στοχαστικής κατάβασης κλίσης που στηρίζεται στον αλγόριθμο της κατάβασης κλίσης, τους αλγορίθμους RMSProp, Adam και Nadam. Συγκεκριμένα, ο αλγόριθμος Adam μοιάζει με τον αλγόριθμο RMSprop και ο αλγόριθμος Nadam με τον αλγόριθμο Adam [13].

Μερικοί από τους λόγους που μας ώθησαν να χρησιμοποιήσουμε αυτούς τους αλγόριθμους βελτιστοποίησης είναι:

- Να μειώσουμε το πλήθος των εποχών εκπαίδευσης των δικτύων μέχρι την απόκτηση ενός εύλογου ποσοστού ακριβείας κατηγοριοποίησης δεδομένων κατά την εκπαίδευση και επαλήθευση του δικτύου.
- Να αποκτήσουμε ψηλότερα ποσοστά ακριβείας κατηγοριοποίησης δεδομένων κατά την εκπαίδευση κα επαλήθευση του δικτύου.

- Να ρυθμίσουμε την συμπεριφορά του δικτύου σύμφωνα με ένα μεγάλο εύρος υπερπαραμέτρων και όχι βάσει του ρυθμού μάθησης [13].

Ρυθμός Μάθησης (Learning Rate)

Ο ρυθμός μάθησης είναι μια σημαντική υπερπαραμέτρος ρύθμισης ενός νευρωνικού δικτύου. Συμπράττει στην μεταβολή των συνοπτικών βαρών ενός νευρωνικού δικτύου. Η επιτυχής εκπαίδευση ενός νευρωνικού δικτύου εξαρτάται από την σωστή διαχείριση της μεταβολής του ρυθμού μάθησης. Στις επόμενες υποενότητες, θα μελετήσουμε τους αλγορίθμους εκπαίδευσης που χρησιμοποιήθηκαν κατά την υλοποίηση του ανιχνευτή αντικειμένων καθώς και ποιες παραμέτρους μπορούμε να ρυθμίσουμε σε αυτούς. Στην συνέχεια, στο κεφάλαιο 6 κάνουμε μερικά πειράματα στα οποία θα συγκρίνουμε τους αλγορίθμους αυτούς όσον αφορά την ανίχνευση και τον εντοπισμό της θέσης των οχημάτων σε μια εικόνα εισόδου [13].

Στοχαστική Κατάβαση Κλίσης

Σε προηγούμενη παράγραφο συζητήσαμε για τον αλγόριθμο κατάβασης κλίσης, έναν αλγόριθμο βελτιστοποίησης πρώτης τάξης που μπορεί να χρησιμοποιηθεί ώστε το δίκτυο να μάθει ένα σύνολο βαρών κατηγοριοποίησης. Αυτή η υλοποίηση όμως εκτελείται πολύ αργά σε μεγάλα σύνολα δεδομένων. Επομένως, χρησιμοποιούμε τον αλγόριθμο στοχαστικής κατάβασης κλίσης που με μια απλή τροποποίηση στον αλγόριθμο κατάβασης κλίσης υπολογίζει την κλίση και ενημερώνει τον πίνακα των βαρών W σε μικρά batches του συνόλου δεδομένων εκπαίδευσης αντί ολόκληρο το σύνολο δεδομένων εκπαίδευσης. Ο αλγόριθμος αυτός μας επιτρέπει να κάνουμε πιο πολλά βήματα πάνω στην κλίση (1 βήμα ανά batch σε αντίθεση με 1 βήμα ανά εποχή στον αλγόριθμο κατάβασης κλίσης), πραγματοποιώντας πιο πολλές ενημερώσεις συναπτικών βαρών και επομένως να μας

οδηγήσει σε γρηγορότερη σύγκλιση χωρίς να επηρεάσει αρνητικά το ποσοστό ακρίβειας κατηγοριοποίησης ή το ποσοστό σφάλματος κατηγοριοποίησης [12, 13].

Ο αλγόριθμος στοχαστικής κατάβασης κλίσης είναι ο πιο σημαντικός αλγόριθμος εκπαίδευσης βαθιών νευρωνικών συνελκτικών δικτύων. Παρόλο που ο αλγόριθμος στοχαστικής κατάβασης κλίσης προτάθηκε πριν 57 χρόνια, εξακολουθεί να είναι βασικός αλγόριθμος στην εκπαίδευση στην εκμάθηση μοτίβων από σημεία δεδομένων [12, 13].

Υπάρχουν δύο σημαντικές υπερπαραμέτροι στον αλγόριθμο Στοχαστικής Κατάβασης Κλίσης. Η πρώτη υπερπαραμέτρος είναι η ορμή που επιταχύνει την στοχαστική κατάβαση κλίσης. Αυτή η υπερπαραμέτρος βοηθάει το δίκτυο να μαθαίνει πιο γρήγορα αφού επικεντρώνεται στις διαστάσεις των οποίων οι κλίσεις δείχνουν προς την ίδια κατεύθυνση. Η δεύτερη υπερπαραμέτρος είναι η επιτάχυνση Nesterov που αποτελεί επέκταση της πρώτης παραμέτρου [13].

Ορμή

Καθώς κατεβαίνουμε μαζί με την κλίση στον αλγόριθμο κατάβασης κλίσης, αποκτούμε περισσότερη ορμή, που μας μεταφέρει πιο γρήγορα σε ένα τοπικό ελάχιστο.

Η υπερπαραμέτρος της ορμής αυξάνει την δύναμη των ενημερώσεων των βαρών των οποίων οι κλίσεις δείχνουν προς την ίδια κατεύθυνση ενώ μειώνει την δύναμη των ενημερώσεων των βαρών των οποίων οι κλίσεις αλλάζουν κατεύθυνση. Έτσι μπορούμε να πούμε πως τα επόμενα βήματα πάνω στην κλίση (επόμενες ενημερώσεις συναπτικών βαρών) γίνονται σύμφωνα με τα προηγούμενα (προηγούμενες ενημερώσεις), αυξάνοντας το ποσοστό ακρίβειας με το πέρασμα λιγότερων εποχών εκπαίδευσης [13].

$$W = W - \alpha \nabla_w f(W)$$

Εξίσωση 3.6 – Εξίσωση Ενημέρωσης των Βαρών που περιλαμβάνει το ρυθμό μάθησης.

$$V = \gamma(V_{t-1}) + \alpha \nabla w f(W)$$

Εξίσωση 3.7 – Εξίσωση Όρου Ορμής (V).

$$W = W - Vt$$

Εξίσωση 3.8 — Εξίσωση Ενημέρωσης των Βαρών που περιλαμβάνει τον Όρο Ορμής (V).

Προστίθεται ο όρος της ορμής (V) στην προηγούμενη εξίσωση, και η παράμετρος g.

Ο όρος της ορμής g συνήθως τίθεται στην τιμή 0.9. Άλλη μια πρακτική είναι η τιμή του g να είναι 0.5 μέχρι να σταθεροποιηθεί η μάθηση και μετά η τιμή του g να γίνει 0.9 [13].

Επιτάχυνση Nesterov

Η επιτάχυνση Nesterov είναι μια σημαντική υπερπαραμέτρος που χρησιμοποιείται για να μην υπερπηδήσουμε κάποιο τοπικό ελάχιστο κατά την κατάβαση κλίσης. Η επιτάχυνση Nesterov μπορεί να γίνει αντιληπτή ως μια διορθωτική ενημέρωση στην ορμή που ξέρει να επιβραδύνει κοντά στα τοπικά ελάχιστα. Όταν χρησιμοποιούμε την υπερπαραμέτρο της ορμής, πρώτα υπολογίζουμε την κλίση και στη συνέχεια πηδούμε προς την κατεύθυνση της κλίσης. Με την χρήση της επιτάχυνσης Nesterov, πρώτα πηδούμε προς την κατεύθυνση της προηγούμενης κλίσης, μετρούμε την κλίση και στη συνέχεια την διορθώνουμε [13].

RMSProp

Ο RMSProp είναι ένας αλγόριθμος βελτιστοποίησης που δεν δημοσιεύθηκε σε κάποιο ερευνητικό κείμενο.

Ο αλγόριθμος αυτός προσπαθεί να συρρικνώσει τις αρνητικές επιπτώσεις της παγκόσμιας συγκέντρωσης cache μετατρέποντας την σε σταθμισμένο μέσο όρο που μετατοπίζεται [13].

Ψευδωκώδικας Ενημέρωσης RMSProp:

$$\text{cache} = \text{decay_rate} * \text{cache} + (1 - \text{decay_rate}) * (\text{dW} ** 2)$$
$$W += -lr * \text{dW} / (\text{np.sqrt}(\text{cache}) + \text{eps})$$

Ο ρυθμός φθοράς r , είναι μια υπερπαραμέτρος της οποίας η τιμή τίθεται συνήθως σε 0.9. Παρατηρούμε πως προηγούμενες εγγραφές στην cache σταθμίζονται με μικρότερα βάρη από ότι οι νέες ενημερώσεις.

Η ιδιότητα του μέσου όρου που μετακινείται στον αλγόριθμο RMSprop επιτρέπει στην cache να αποδεσμεύει παλιές κλίσεις που είναι τετραγωνισμένες και να τις αντικαταστήσει με νέες [13].

Η ουσία του αλγορίθμου εξαρτάται στην εκθετική φθορά της cache, βοηθώντας μας να αποφεύγουμε μονοτονικούς φθίνοντες ρυθμούς μάθησης κατά την εκπαίδευση του δικτύου. Στην πράξη, ο αλγόριθμος RMSprop είναι πιο αποτελεσματικός άλλους αλγόριθμους (τον Adagrad, Adadelta) όταν εφαρμόζεται κατά την εκπαίδευση ποικίλων αρχιτεκτονικών βαθιών νευρωνικών δικτύων. Επίσης, ο αλγόριθμος RMSprop συγκλίνει πιο γρήγορα από τον SGD. Επιπρόσθετα, είναι ο δεύτερος πιο χρησιμοποιημένος αλγόριθμος βελτιστοποίησης στην βιβλιογραφία της βαθιάς μάθησης. Παρόλα αυτά, ο αλγόριθμος Adam, στον οποίο θα αναφερθούμε πιο κάτω, τώρα χρησιμοποιείται περισσότερο από τον RMSprop [13].

Adam

Ο αλγόριθμος βελτιστοποίησης Adam (Adaptive Moment Estimation προτάθηκε από τους Kingma και Ba το 2014. Ο αλγόριθμος Adam λειτουργεί ακριβώς όπως τον αλγόριθμο RMSprop και προστίθεται σε αυτόν η παράμετρος της ορμής [13].

Ψευδοκώδικας Ενημέρωσης Adam:

$$v = \text{beta2} * v + (1 - \text{beta2}) * (dW ** 2)$$

$$W += -lr * m / (\text{np.sqrt}(v) + \text{eps})$$

Οι τιμές των παραμέτρων m και v εξαρτώνται από τις προηγούμενες τους τιμές στην χρονική στιγμή $t-1$.

Η τιμή της παραμέτρου m αναπαριστά τον μέσο των κλίσεων και η παράμετρος v . Η Ενημέρωση των βαρών που συμβαίνει στον Adam είναι σχεδόν η ίδια με αυτήν που συμβαίνει στον RMSprop, με την διαφορά πως χρησιμοποιούμε μια λεία έκδοση (επειδή υπολογίζουμε τον μέσο) του m αντί της κλίσης dW . Λαμβάνουμε πιο επιθυμητές ενημερώσεις βαρών όταν χρησιμοποιούμε τον μέσο διότι μπορούμε να λειάνουμε τις θορυβώδεις ενημερώσεις στις τιμές της κλίσης dW [13].

Συνήθως, η τιμή της παραμέτρου beta1 τίθεται σε 0.9 ενώ η τιμή της παραμέτρου beta2 σε 0.999. Οι τιμές των παραμέτρων που έχουμε προαναφέρει αλλάζουν σπάνια όταν χρησιμοποιούμε τον αλγόριθμο βελτιστοποίησης Adam. Στην πράξη ο Adam δουλεύει καλύτερα από τον RMSprop σε αρκετές περιπτώσεις [13].

Nadam

Ο αλγόριθμος Nadam λειτουργεί ακριβώς όπως ο RMSprop αλλά με την προσθήκη της Επιτάχυνσης Nesterov. Ο αλγόριθμος αυτός προτάθηκε από τον διδακτορικό φοιτητή του Πανεπιστημίου Stanford Timothy Dozat. Η συγκεκριμένη παραλλαγή του αλγορίθμου Adam χρησιμοποιείται σπάνια [13].

Logistic Regression Classifier

Ένα δίκτυο κατηγοριοποίησης οχημάτων σε εναέριες λήψεις μαθαίνει να ταξινομεί τις εικόνες σε αυτές που περιέχουν οχήματα και σε αυτές που δεν περιέχουν. Κατά την διαδικασία της εκπαίδευσης, το δίκτυο μαθαίνει με την χρήση ενός αλγορίθμου μάθησης (στην παρούσα διπλωματική εργασία χρησιμοποιούμε τον αλγόριθμο Logistic Regression για κάθε δίκτυο), μεταβάλλοντας τα βάρη που διαθέτει καθώς βλέπει καινούργια δείγματα εικόνων [12, 13].

Το Logistic Regression, ψάχνει τις καλύτερες τιμές συγκεκριμένων παραμέτρων για τον εκτιμητή μας. Στις εφαρμογές μας ψάχνουμε τις καλύτερες τιμές (και cross-validation) της παραμέτρου C για τον εκτιμητή Logistic Regression [12,13].

3.3.1.1.2 Αρχιτεκτονικές Βαθιών Συνελικτικών Δικτύων που Χρησιμοποιήθηκαν

Τα βαθιά συνελικτικά νευρωνικά δίκτυα μοιάζουν πολύ με τα κλασσικά τεχνητά νευρωνικά δίκτυα εμπρόσθιου περάσματος και αποτελούν εξέλιξη τους. Ένα συνελικτικό δίκτυο απαρτίζεται από ένα ή περισσότερα συνελικτικά επίπεδα. Πρώτα υπάρχει ένα βήμα υποδειγματοληψίας και ακολούθως ένα ή περισσότερα πλήρως συνδεδεμένα στρώματα (όπως στα παραδοσιακά πολυστρωματικά νευρωνικά δίκτυα. Η αρχιτεκτονική του συνελικτικού δικτύου σχεδιάστηκε με τέτοιο τρόπο που να μπορεί να εκμεταλλευτεί τα χαρακτηριστικά μιας τρισδιάστατης εικόνας εισόδου ή ενός δυσδιάστατου ηχητικού σήματος ως είσοδο. Αυτό μπορεί να συμβεί με την χρήση συνελικτικών επιπέδων που ακολουθούνται από επίπεδα συγκέντρωσης βοηθάει στην μετάφραση συγκεκριμένων αμετάβλητων χαρακτηριστικών των δεδομένων. Ένα άλλο πλεονέκτημα των συνελικτικών δικτύων είναι ότι αυτά έχουν αραιή συνδεσιμότητα μεταξύ των νευρώνων του δικτύου σε

σύγκριση με τα πλήρως συνδεδεμένα δίκτυα εμπρόσθιου περάσματος που έχουν το ίδιο πλήθος νευρώνων. Συγκεκριμένα, ένας κρυφός νευρώνας στο κρυφό επίπεδο είναι συνδεδεμένος με μια μικρή περιοχή των νευρώνων n του προηγούμενου επιπέδου. Ακόμη, μια σημαντική λειτουργία του συνελκτικού δικτύου είναι η λειτουργία μιας συνέλιξης και είναι χρήσιμη στην επεξεργασία εικόνας. Η χρήση μιας λειτουργίας συνέλιξης, δίνει την δυνατότητα στο δίκτυο να επικεντρωθεί στα σημαντικά χαρακτηριστικά των δεδομένων εισόδου και να το κατευθύνει ώστε να εκπαιδευτεί για να επιλύει το πρόβλημα που μας ενδιαφέρει. Παραδείγματος χάριν, η εφαρμογή των συνελκτικών δικτύων για την επίλυση προβλημάτων που αφορούν αναγνώριση αντικειμένων σε εικόνες θα ήταν πολύ χρήσιμη. Οι εικόνες περιέχουν περιττές πληροφορίες που δεν είναι χρήσιμες κατά την αναγνώριση αντικειμένων σε εικόνες. Η σημαντικότερη πληροφορία κατά την αναγνώριση αντικειμένων είναι οι ακμές. Επομένως, κατασκευάζουμε ένα μηχανισμό ανίχνευσης ακμών με την χρήση των συνελιξεων ενός συνελκτικού δικτύου για να επιλύσουμε το πρόβλημα αυτό. Ο κύριος λόγος για τον οποίο χρησιμοποιούμε συνελκτικά νευρωνικά δίκτυα στις εργασίες μάθησης της μηχανικής όρασης είναι η χρήση τοπικών προτύπων και οι αρχιτεκτονική τους που έχει δημιουργηθεί με στόχο να λύνει προβλήματα στα οποία τα δεδομένα (ήχος, εικόνα, βίντεο) έχουν την έννοια της συνέχειας [5].

*Συνέχεια Δεδομένων: Τα δεδομένα μπορούν να χωριστούν και να συνεχίσουν να έχουν μια στενή σχέση με τα προηγούμενα και τα επόμενα πρότυπα εισόδων. Τα πρότυπα χωρίζονται με την χρήση ενός μετακινούμενου παράθυρου.

Τα βαθιά συνελκτικά νευρωνικά δίκτυα μοιάζουν πολύ με τα κλασσικά τεχνητά νευρωνικά δίκτυα εμπρόσθιου περάσματος και αποτελούν εξέλιξη τους. Απαρτίζονται από νευρώνες οι οποίοι έχουν βάρη και κατώφλια που προσαρμόζονται κατά την εκπαίδευση. Κάθε νευρώνας λαμβάνει μια είσοδο, υπολογίζει το εσωτερικό γινόμενο και εισάγει το αποτέλεσμα του σε μια μη γραμμική συνάρτηση ενεργοποίησης. Επίσης, υπάρχουν συναρτήσεις υπολογισμού του σφάλματος στο τελευταίο επίπεδο εξόδου. Όπως προαναφέραμε τα κλασσικά τεχνητά νευρωνικά δίκτυα, λαμβάνουν ένα διάνυσμα εισόδου, το περνούν μέσα από μια σειρά κρυφών επιπέδων (διαφορικών εξισώσεων). Ένα

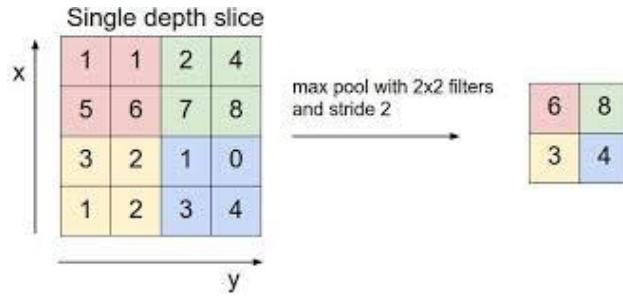
επίπεδο του δικτύου απαρτίζεται από πολλούς νευρώνες όπου κάθε νευρώνας είναι πλήρως συνδεδεμένος με όλους τους νευρώνες του προηγούμενου επιπέδου. Οι νευρώνες που βρίσκονται στο ίδιο επίπεδο ενός νευρωνικού δικτύου δεν έχουν μεταξύ τους συναπτικά βάρη που να τους συνδέουν και δρουν ανεξάρτητα. Το τελευταίο επίπεδο του νευρωνικού δικτύου ονομάζεται και επίπεδο εξόδου. Κατά την επίλυση προβλημάτων κατηγοριοποίησης δείχνει την κατηγορία στην οποία ανήκει το δεδομένο ενώ κατά την επίλυση προβλημάτων πρόβλεψης παρουσιάζει την πρόβλεψη του [5].

Τα πρώτα επίπεδα σε ένα συνελικτικό δίκτυο είναι επίπεδα συνελίξεων (convolutional layers, τα αμέσως επόμενα επίπεδα είναι επίπεδα συγκέντρωσης (pooling layers). Τα τελευταία επίπεδα σε μια τέτοια αρχιτεκτονική δικτύων είναι συνήθως πλήρως συνδεδεμένα επίπεδα. Στα εν λόγω επίπεδα οι νευρώνες του επιπέδου i συνδέονται με μια ακμή με κάθε νευρώνα του επιπέδου j και οι νευρώνες του επιπέδου j συνδέονται με μια ακμή με κάθε νευρώνα του επιπέδου i [5].

Ένα συνελικτικό δίκτυο διαθέτει συνελικτικά επίπεδα, υποδειγματοληπτικά επίπεδα και πλήρως συνδεδεμένα επίπεδα. Η είσοδος ενός συνελικτικού επιπέδου είναι μια εικόνα μεγέθους $m \times n \times r$, όπου το m είναι το ύψος και το πλάτος της εικόνας και το r το πλήθος των καναλιών της. Για παράδειγμα, μια εικόνα στον χώρο αναπαράστασης Red Green Blue (RGB) όπως εξηγήσαμε και σε προηγούμενη παράγραφο, διαθέτει πλήθος καναλιών $r = 3$, ένα κανάλι για κάθε συστατικό/χρώμα στον χώρο αναπαράστασης της. Ένα επίπεδο συνέλιξης διαθέτει k kernels (φίλτρα) μεγέθους $n \times n \times q$ όπου το n είναι μια τιμή μικρότερη από αυτήν της m (διαστάσεων της εικόνας) και μια τιμή q που μπορεί να είναι ίση με το πλήθος των καναλιών r ή μικρότερο αυτού και επίσης η τιμή q μπορεί να είναι διαφορετική για σε κάθε φίλτρο (kernel). Τα φίλτρα συνελίσσονται με την είσοδο το δικτύου, δημιουργώντας k χάρτες χαρακτηριστικών (feature maps) μεγέθους $m-n+1$. Το επίπεδο συγκέντρωσης ή υποδειγματοληψίας τοποθετείται προαιρετικά σε ένα συνελικτικό δίκτυο. Εμείς θα το τοποθετήσουμε καθώς είναι απαραίτητο στις εφαρμογές που θα υλοποιήσουμε. Ακολούθως, πραγματοποιείται υποδειγματοληψία σε κάθε χάρτη

χαρακτηριστικών με τυπική ή μέγιστη συγκέντρωση πάνω από τις ίδιες περιοχές στους χάρτες χαρακτηριστικών μεγέθους $\rho \times \rho$ όπου το ρ είναι μια τιμή που κυμαίνεται μεταξύ του 2 και του 5. Όσο πιο μικρό είναι το μέγεθος της εισόδου τόσο πιο μικρή τιμή παίρνει τη παράμετρος ρ . Πριν ή μετά το επίπεδο συγκέντρωσης ή υποδειγματοληψίας εφαρμόζεται ένα bias και μια μη γραμμική συνάρτηση σε κάθε χάρτη χαρακτηριστικών (feature map) [5].

Σε μια κλασσική αρχιτεκτονική συνελκτικών δικτύων, μια συνηθισμένη τακτική είναι να προσθέτουμε ένα επίπεδο συγκέντρωσης ή υποδειγματοληψίας μεταξύ κρυφών συνελκτικών επιπέδων. Ο λόγος για τον οποίο χρησιμοποιούμε επίπεδα συγκέντρωσης ή επίπεδα υποδειγματοληψίας είναι για να μειώσουμε το Ο κύριος λόγος για την χρήση pooling layers, είναι για να μειώσουμε μη γραμμικά τις διαστάσεις (το μέγεθος της αναπαράστασης των ενδιάμεσων αποτελεσμάτων), το πλήθος των ελεύθερων παραμέτρων και την πολυπλοκότητα του δικτύου. Με την χρήση των πιο πάνω πρακτικών, το δίκτυο μας δεν υπερεκπαιδεύεται. Τα επίπεδα υποδειγματοληψίας επεξεργάζονται κάθε έξοδο που προέρχεται από κάθε kernel ξεχωριστά και δρουν ανεξάρτητα από τα άλλα επίπεδα. Συνήθως, ένα επίπεδο υποδειγματοληψίας εφαρμόζει φίλτρα μεγέθους 2×2 , με stride 2 και padding 2. Η συνηθισμένη τεχνική υποδειγματοληψίας που χρησιμοποιεί είναι η downsampling είναι η μέγιστη. Σε κάθε φιλτραρισμένο αποτέλεσμα εξόδου, η προαναφερθείσα τεχνική, επιλέγει τον μεγαλύτερο όρο από το σύνολο των όρων που εξετάζει. Για παράδειγμα, όσον αφορά μια είσοδο μεγέθους $2 \times 2 = 4$, επιλέγει τον μεγαλύτερο από τους 4 όρους. σε κάθε φιλτραρισμένο αποτέλεσμα εξόδου. Επιπρόσθετα, οι μονάδες συγκέντρωσης έχουν την δυνατότητα να χρησιμοποιήσουν συναρτήσεις όπως L2 συγκέντρωση κανονικοποίησης, συγκέντρωση μέσου όρου κλπ. Η τεχνική μέγιστη (max) δίνει τα καλύτερα αποτελέσματα [5].



Σχήμα 3.7 – Υποδειγματοληψία/Downsampling

*Υπερεκπαίδευση συμβαίνει όταν το σφάλμα εκπαίδευσης μειώνεται ενώ τα σφάλμα επαλήθευσης αυξάνεται.

Επίπεδο Συνέλιξης.

Ένα επίπεδο συνέλιξης αποτελείται από πολλά φίλτρα. Στο εμπρόσθιο πέρασμα, κάθε φίλτρο συνελίσσεται κατά μήκος του πλάτους και του ύψους του όγκου διανύσματος εισόδου. Εκτελείται ένα εσωτερικό γινόμενο μεταξύ του φίλτρου και του διανύσματος εισόδου. Ακολουθώντας, δημιουργείται ένας χάρτης 2 διαστάσεων ενεργοποίησης που αφορά το συγκεκριμένο φίλτρο. Η διαδικασία αυτή έχει ως αποτέλεσμα το δίκτυο να μαθαίνει φίλτρα που ενεργοποιούνται σε ακμές ή χρώματα στο πρώτο επίπεδο συνέλιξης και σε σχήματα ή αντικείμενα στα επόμενα επίπεδα συνέλιξης. Το επίπεδο συνέλιξης δέχεται δεδομένα μεγέθους $W_i \times H_i \times D_i$. Λαμβάνει 4 παραμέτρους, το Stride D , το Padding P , το πλήθος των φίλτρων K και το μέγεθος του κάθε φίλτρου. Παράγει αποτέλεσμα μεγέθους $W_j \times H_j \times D_j$, όπου:

$$W_j = \frac{W_i - F + 2P}{S + 1}$$

Εξίσωση 3.19 — Εξίσωση του Πλάτους στο Επίπεδο Συνέλιξης j σύμφωνα με τις διαστάσεις των δεδομένων εισόδου στο επίπεδο εισόδου i .

$$H_j = \frac{H_i - F + 2P}{S + 1}$$

Εξίσωση 3.10 — Εξίσωση του Μήκους H στο Επίπεδο Συνέλιξης j σύμφωνα με τις διαστάσεις των δεδομένων εισόδου στο επίπεδο εισόδου i .

$$D_j = K$$

Εξίσωση 3.11 — Εξίσωση του Βάθους D στο Επίπεδο Συνέλιξης j σύμφωνα με τις διαστάσεις των δεδομένων εισόδου στο επίπεδο εισόδου i .

Για κάθε φίλτρο παράγονται $F \times F \times D_i + 1$ βάρη και συνολικά $F \times F \times D_1 \times K$ βάρη και K biases [1].

Επίπεδο Συγκέντρωσης

Ένα επίπεδο συγκέντρωσης μειώνει μη γραμμικά τις διαστάσεις των δεδομένων εισόδου, τις ελεύθερες παραμέτρους και την πολυπλοκότητα του δικτύου. Η χρήση αυτού του επιπέδου μας βοηθάει να αποφύγουμε το φαινόμενο της υπερεκπαίδευσης που συμβαίνει κατά την εκπαίδευση του δικτύου. Η συνήθης συνάρτηση που χρησιμοποιείται στο επίπεδο συγκέντρωσης είναι η μέγιστη. Αυτό οφείλεται στα καλά αποτελέσματα που παράγει η εν λόγω συνάρτηση. Το επίπεδο συγκέντρωσης δέχεται δεδομένα μεγέθους $W_i \times H_i \times D_i$. Λαμβάνει 2 υπερπαραμέτρους, το Stride S και την χωρική έκταση F . Παράγει αποτέλεσμα μεγέθους $W_j \times H_j \times D_j$, όπου [1]:

$$W_j = \frac{W_i - F}{S + 1}$$

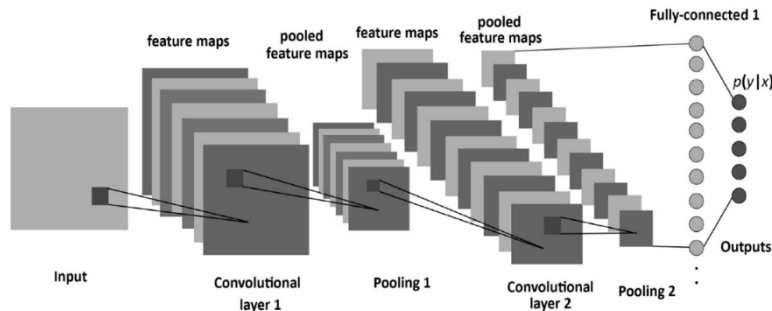
Εξίσωση 3.12 — Εξίσωση του Πλάτους W στο Επίπεδο Συγκέντρωσης j σύμφωνα με τις διαστάσεις των δεδομένων εισόδου στο επίπεδο εισόδου i .

$$H_j = \frac{H_i - F}{S + 1}$$

Εξίσωση 3.13 — Εξίσωση του Μήκους H στο Επίπεδο Συγκέντρωσης j σύμφωνα με τις διαστάσεις των δεδομένων εισόδου στο επίπεδο εισόδου i .

$$D_j = D_i$$

Εξίσωση 3.14 — Εξίσωση του Βάθους D στο Επίπεδο Συγκέντρωσης j σύμφωνα με τις διαστάσεις των δεδομένων εισόδου στο επίπεδο εισόδου i .



Σχήμα 3.8– Αναπαράσταση Αρχιτεκτονικής Επιπέδων Συνελικτικών Δικτύων

Αρχιτεκτονική Συνελικτικού Δικτύου ResNet50

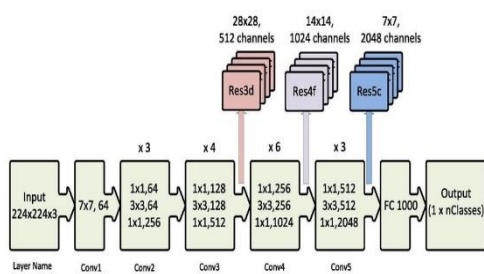
Το 2015, το βαθύ συνελικτικό δίκτυο ResNet ήταν ο νικητής του διαγωνισμού ILSVRC. Το δίκτυο αυτό περιέχει συγκεκριμένες συνδέσεις που ονομάζονται “skip connections” και χρησιμοποιεί πάρα πολύ το batch normalization. Οι συνδέσεις που προαναφέραμε αναφέρονται στην βιβλιογραφία ως residual connections ή gated recurrent units και μοιάζουν πάρα πολύ με στοιχεία που εφαρμόζονται στα δίκτυα RNN [1, 2, 4].

Επιπρόσθετα, η αρχιτεκτονική αυτή είναι πιο απλή από την αρχιτεκτονική VGGNet, δεν διαθέτει πλήρως συνδεδεμένα επίπεδα στα τελευταία της επίπεδά, ενώ αποτελείται από 152 επίπεδα συνέλιξης και 176 επίπεδα συνολικά. RNN [4]. Το ResNet είναι πολύ δημοφιλές μέχρι σήμερα και είναι η πρώτη επιλογή στις πρακτικές εφαρμογές βαθιών συνελικτικών δικτύων. Αυτό οφείλεται στο πολύ χαμηλό ποσοστό σφάλματος που πετυχαίνει σε πολλές εφαρμογές και είναι χαμηλότερο από αυτό που θα έκανε κάποιος άνθρωπος καθώς θα εκτελούσε μια εργασία [4, 10].

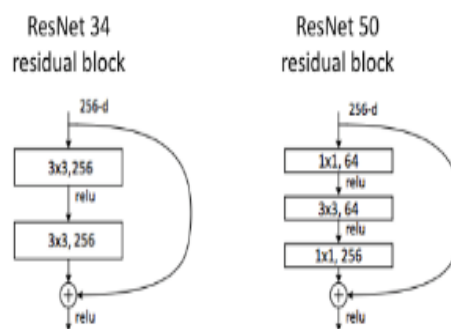
Το δίκτυο ResNet50 που χρησιμοποιήσαμε περιέχει 25 million παραμέτρους. Κάθε χρωματιστό μπλοκ επιπέδων αναπαριστά σειρές συνελίξεων της ίδιας διάστασης. Το μπλοκ της χαρτογράφησης των χαρακτηριστικών εκτελεί υποδειγματοληψία με την χρήση strided συνέλιξης και μια αύξηση του βάθους των καναλιών για να κρατήσει την πολυπλοκότητα του χρόνου ανά επίπεδο.

Η αρχιτεκτονική του δικτύου περιλαμβάνει ένα τριών επιπέδων bottleneck μπλοκ που χρησιμοποιεί συνελίξεις μεγέθους 1x1 για να μειώσει και να επαναφέρει το βάθος των καναλιών, μειώνοντας τον υπολογιστικό φόρτο κατά τον υπολογισμό συνελίξεων μεγέθους 3x3 [1, 2, 4, 7, 10].

Οι ελάχιστες διαστάσεις των δεδομένων εισόδου του ResNet είναι 197x197 pixels ενώ οι προκαθορισμένες είναι 224x224 pixels. Στις εφαρμογές που υλοποιήσαμε χρησιμοποιήσαμε τις προκαθορισμένες διαστάσεις δεδομένων εισόδου καθώς και τις ελάχιστες δυνατές διαστάσεις (197x197 pixels).



Σχήμα 3.9 - Αρχιτεκτονική Δικτύου ResNet50



Σχήμα 3.10 - Residual Block ResNet34 vs ResNet50

Αρχιτεκτονική Συνελικτικού Δικτύου VGG16

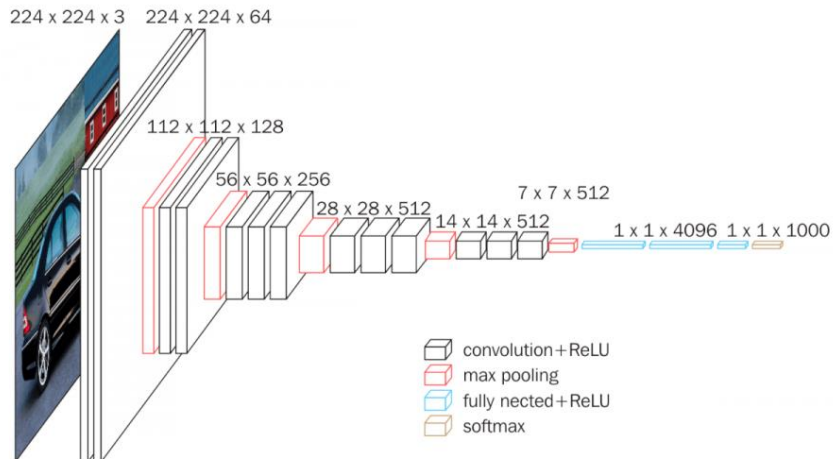
Το δίκτυο VGG16 ανακαλύφθηκε το 2014, αποτελείται από 22 επίπεδα συνολικά και ανήκει στις κλασικές αρχιτεκτονικές συνελικτικών δικτύων. Εκπαιδεύεται συνήθως για 2-3 εβδομάδες χρησιμοποιώντας 4 GPUS και χρησιμοποιείται συχνά για την εξαγωγή

χαρακτηριστικών από εικόνες. Η εφεύρεση του VGGNet απόδειξε ότι το βάθος ενός νευρωνικού συνελκτικού δικτύου συνδέεται άμεσα με την καλή του επίδοση. Το VGG16 αποτελείται 16 επίπεδα συνελίξεων (convolutional layers) που εφαρμόζουν συνελίξεις μεγέθους 3x3 με stride 1 και pad 1 και επίπεδα συγκέντρωσης (pooling layers) που εκτελούν μέγιστη συγκέντρωση μεγέθους 2x2 με stride 2 και χωρίς padding [1, 2, 4, 10].

Ακόμη, διαθέτει πάρα πολλές παραμέτρους που οι περισσότερες βρίσκονται στο πρώτο πλήρως συνδεδεμένο επίπεδο συνέλιξης. Επομένως, είναι πολύ ακριβό στην αξιολόγηση αφού χρησιμοποιεί περισσότερη μνήμη σε σχέση με τα άλλα δίκτυα για να αποθηκεύσει τις 140 million παραμέτρους του. Αργότερα ανακαλύφθηκε πως μπορούμε να αφαιρέσουμε τα πρώτα επίπεδα του χωρίς να επηρεαστεί αρνητικά η απόδοση του δικτύου ενώ παράλληλα μειώνεται το πλήθος των παραμέτρων του δικτύου [2, 10].

Η αρχιτεκτονική του δικτύου VGG16 αποτελείται από 16 επίπεδα συνέλιξης που το καθένα από αυτά ακολουθείται από ένα RELU επίπεδο και στο τέλος υπάρχει ένα επίπεδο συγκέντρωσης. Αυτή η αρχιτεκτονική επιτρέπει στο δίκτυο να μαθαίνει σύνθετα χαρακτηριστικά στα επίπεδα συνέλιξης πριν την υποδειγματοληψία του χωρικού μεγέθους της εισόδου που λαμβάνει χώρα στα επίπεδα συγκέντρωσης [12].

Οι ελάχιστες διαστάσεις των δεδομένων εισόδου του VGG16 είναι 32x32 pixels ενώ οι προκαθορισμένες είναι 224x224 pixels. Στις εφαρμογές που υλοποιήσαμε χρησιμοποιήσαμε τις προκαθορισμένες διαστάσεις δεδομένων εισόδου (στο VGG16 και στο VGG16 Transfer Learning) καθώς και μικρότερες διαστάσεις (VGG16 & VGG16 Transfer Learning: 192x192 pixels).



Σχήμα 3.11 - Αρχιτεκτονική Δικτύου VGG16 - Input Shape: 224x224 pixels Image

Αρχιτεκτονική Συνελικτικού Δικτύου ShallowNet

Το δίκτυο ονομάζεται ShallowNet αφού η αρχιτεκτονική του αποτελείται από λιγότερα επίπεδα σε σύγκριση με άλλες αρχιτεκτονικές συνελικτικών δικτύων. Χρησιμοποιεί μόνο ένα επίπεδο συνέλιξης που ακολουθείται από ένα επίπεδο RELU. Το επίπεδο RELU ακολουθείται από ένα πλήρως συνδεδεμένο επίπεδο (FC). Το ποσοστό ακριβείας ενός δικτύου αυξάνεται όσο προστίθενται περισσότερα σύνολα πράξεων: συνέλιξης->RELU->συγκέντρωσης. Αυτή η αρχιτεκτονική όταν εκπαιδευτεί στο σύνολο δεδομένων CIFAR-10 και στο σύνολο δεδομένων Animals δίνει καλά αποτελέσματα, συγκεκριμένα 60% ποσοστό ακριβείας στην κατηγοριοποίηση CIFAR-10 αντικειμένων και 71% ποσοστό ακριβείας στην κατηγοριοποίηση ζώων. Οι ελάχιστες διαστάσεις των δεδομένων εισόδου του ShallowNet είναι 32x32 pixels. Στις εφαρμογές που υλοποιήσαμε χρησιμοποιήσαμε τις ελάχιστες διαστάσεις δεδομένων εισόδου [12].

ShallowNet 64 X 64		ShallowNet 128 X 128	
Layer (type)	Output Shape	Layer (type)	Output Shape
conv2d_19 (Conv2D)	(None, 64, 64, 32)	conv2d_20 (Conv2D)	(None, 128, 128, 32)
activation_10 (Activation)	(None, 64, 64, 32)	activation_12 (Activation)	(None, 128, 128, 32)
max_pooling2d_20 (MaxPooling)	(None, 32, 32, 32)	max_pooling2d_22 (MaxPooling)	(None, 64, 64, 32)
max_pooling2d_21 (MaxPooling)	(None, 16, 16, 32)	max_pooling2d_23 (MaxPooling)	(None, 32, 32, 32)
flatten_7 (Flatten)	(None, 8192)	max_pooling2d_24 (MaxPooling)	(None, 16, 16, 32)
dense_15 (Dense)	(None, 4000)	flatten_8 (Flatten)	(None, 8192)
dense_16 (Dense)	(None, 1)	dense_17 (Dense)	(None, 4000)
activation_11 (Activation)	(None, 1)	dense_18 (Dense)	(None, 1)
		activation_13 (Activation)	(None, 1)

Σχήμα 3.12 Αρχιτεκτονική Δικτύου ShallowNet

Αρχιτεκτονική Συνελικτικού Δικτύου MobileNet

Η αρχιτεκτονική MobileNet χρησιμοποιεί διαχωρισμένες depthwise συνελίξεις. Μια depthwise συνέλιξη εφαρμόζει ένα φίλτρο σε κάθε κανάλι εισόδου. Έπειτα, η pointwise συνέλιξη εφαρμόζει μια συνέλιξη μεγέθους 1x1 για να συνδυάσει τις εξόδους της depthwise συνέλιξης. Μια κανονική συνέλιξη φιλτράρει και συνδυάζει τις εισόδους σε ένα νέο σύνολο εξόδων σε ένα βήμα. Μια ξεχωριστή depthwise συνέλιξη χωρίζει αυτή την διαδικασία σε δύο επίπεδα, ένα ξεχωριστό επίπεδο για φιλτράρισμα και ένα άλλο για τον συνδυασμό των εξόδων. Αυτή η παραγοντοποίηση έχει ως στόχο την μείωση του υπολογιστικού κόστους και του μεγέθους του μοντέλου μέσω της μείωσης των παραμέτρων του δικτύου. Το MobileNet αποτελείται από 92 επίπεδα συνολικά. Το πρώτο επίπεδο του δικτύου χρησιμοποιεί μια πλήρη συνέλιξη με stride 2. Έπειτα, το επόμενο επίπεδο είναι ένα depthwise επίπεδο και το μεθεπόμενο ένα pointwise επίπεδο που διπλασιάζει τον αριθμό των καναλιών. Στην συνέχεια περιέχει τον συνδυασμό depthwise και sterwise επιπέδων πολλές φορές. Ακολούθως, το αποτέλεσμα των πράξεων περνά από ένα μέσο επίπεδο συγκέντρωσης. Το επίπεδο συγκέντρωσης ακολουθείται από ένα πλήρες συνδεδεμένο επίπεδο. Το τελικό πλήρες συνδεδεμένο επίπεδο είναι μικρότερο σε

σχέση με αυτών των κλασικών αρχιτεκτονικών διότι πρώτα εφαρμόζεται global μέγιστη συγκέντρωση.

Οι ελάχιστες διαστάσεις των δεδομένων εισόδου του MobileNet είναι 32x32 pixels ενώ οι προκαθορισμένες είναι 224x224 pixels. Στις εφαρμογές που υλοποιήσαμε χρησιμοποιήσαμε τις προκαθορισμένες διαστάσεις δεδομένων εισόδου (τόσο στο MobileNet όσο και στο MobileNet Transfer Learning) καθώς και μικρότερες διαστάσεις (MobileNet & MobileNet Transfer Learning: 192x192 pixels, MobileNet Transfer Learning: 160x160 pixels, MobileNet Transfer Learning: 128x128 pixels). Το προαναφερθέν δίκτυο είναι πολύ γρήγορο στην σύγκλιση και παρέχει ψηλά ποσοστά ακριβείας [3, 4, 8].

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5 \times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Σχήμα 3.13 Αρχιτεκτονική Δικτύου MobileNet

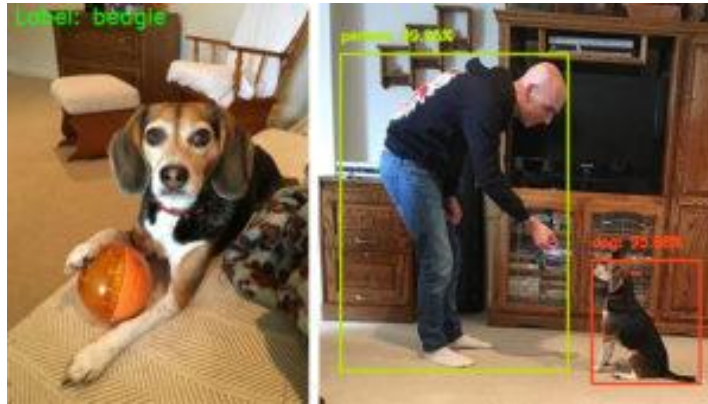
3.6 Αναγνώριση Αντικειμένων vs Ανίχνευση Αντικειμένων vs Κατηγοριοποίηση στην Μηχανική Μάθηση και Μηχανική Όραση

Η εργασία της αναγνώρισης ενός αντικειμένου είναι η αντιστοίχιση ενός συγκεκριμένου αντικειμένου σε μια εικόνα. Μερικά αντικείμενα που μπορούν να αναγνωριστούν είναι μια συγκεκριμένη μάρκα αυτοκινήτου π.χ. Toyota, ένα συγκεκριμένο σήμα τροχαίας π.χ.

ανώτατο όριο ταχύτητας 80 ή μια συγκεκριμένη γέφυρα π.χ της γέφυρας Ponte Vecchio που βρίσκεται στην Φλωρεντία. Ένα αντικείμενο μπορεί να αναγνωρισθεί παρά τις αλλαγές στην κλίμακα απεικόνισης, στις συνθήκες φωτισμού, στην γωνία λήψης και ακόμη και αν ένα μέρος του είναι κρυμμένο. Επομένως, αυτό μας οδηγεί στο συμπέρασμα πως συγκεκριμένα αντικείμενα έχουν την δυνατότητα να περιγράψουν επακριβώς. Άρα, όταν θέλουμε να αντιστοιχίσουμε καινούργια αντικείμενα με τα προυπάρχοντα τότε τα καινούργια αντικείμενα χρειάζεται να έχουν ένα ελάχιστο πλήθος κοινών χαρακτηριστικών με τα προυπάρχοντα για να ανήκουν στην ίδια κατηγορία με αυτά. Αν ένα αντικείμενο δεν ξεπερνά το ελάχιστο πλήθος κοινών χαρακτηριστικών που χρειάζεται να έχει για να ανήκει στην ίδια κατηγορία με κάποιο άλλο τότε δεν αντιστοιχείται [9, 11].

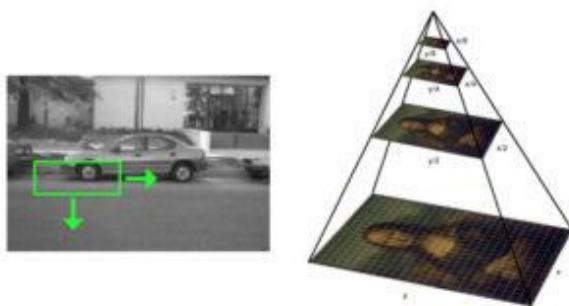
Η εργασία της ανίχνευσης ενός αντικειμένου στοχεύει στην εύρεση ενός αντικειμένου με την μάθηση μιας πιο εξιδεικευμένης περιγραφής του αντικειμένου και όχι μια απλή περιγραφή της εικόνας. Σε περίπτωση που το αντικείμενο που θέλουμε να ανιχνεύσουμε είναι πολύπλοκο (η πολυπλοκότητα ενός αντικειμένου αυξάνεται όσο αυξάνεται η ποικιλομορφία των εικόνων και το πλήθος τους) τότε θα ανιχνεύσουμε μια κλάση αντικειμένων που αντιπροσωπεύει τα αντικείμενα αυτά [9].

Στην κατηγοριοποίηση, ολόκληρη η εικόνα κατηγοριοποιείται με μια ετικέτα. Στην περίπτωση της αναγνώρισης αντικειμένων, ανιχνεύονται πολλαπλά αντικείμενα μέσα στην εικόνα. Στην κλασική κατηγοριοποίηση δοθείσας μιας εικόνας ως δεδομένο εισόδου, την παρουσιάζουμε στο νευρωνικό μας δίκτυο και λαμβάνουμε μια ετικέτα που καθορίζει την κλάση στην οποία ανήκει και ίσως μια ετικέτα που σχετίζεται με αυτή την ετικέτα. Η ετικέτα χαρακτηρίζει τα περιεχόμενα ολόκληρης της εικόνας ή το κυριότερο αντικείμενο που είναι ορατό μέσα στην εικόνα. Επομένως, στην κατηγοριοποίηση μιας εικόνας δίνουμε μια εικόνα εισόδου και λαμβάνουμε μια ετικέτα στην έξοδο [9, 11].



Σχήμα 3.14 – Κατηγοριοποίηση **Σχήμα 3.15 – Ανίχνευση**

Η αναγνώριση και ανίχνευση αντικειμένων μπορεί να πραγματοποιηθεί με την χρήση βαθιών συνελκτικών δικτύων ή άλλων τεχνικών της μηχανικής όρασης. Η ανίχνευση και αναγνώριση αντικειμένων χρησιμοποιεί ως θεμέλιο της την κατηγοριοποίηση και προσπαθεί να εντοπίσει την ακριβή τοποθεσία κάθε αντικειμένου μέσα στην εικόνα εισόδου. Όταν πραγματοποιούμε την αναγνώριση και ανίχνευση αντικειμένων, δοθείσας μιας εικόνας εισόδου, θέλουμε να λάβουμε στην έξοδο κουτιά τα οποία να περικλείουν τα αντικείμενα που μας ενδιαφέρουν ή καρτεσιανές συντεταγμένες (x,y) για κάθε αντικείμενο ενδιαφέροντος που βρίσκεται μέσα στην εικόνα εισόδου, μια ετικέτα που να δηλώνει την κλάση στην οποία ανήκει το κάθε αντικείμενο ενδιαφέροντος (μια ετικέτα ανά κουτί) και μια πιθανότητα ή σκορ εμπιστοσύνης που να σχετίζεται με κάθε κουτί και ετικέτα. Επομένως, στην ανίχνευση και αναγνώριση αντικειμένων, δίνουμε μια εικόνα εισόδου και λαμβάνουμε μια πολλές ετικέτες και πολλά κουτιά στην έξοδο στην έξοδο [11].



Σχήμα 3.16 Sliding Window & Image Pyramid

Αφού μάθαμε την διαφορά μεταξύ κατηγοριοποίησης εικόνων και ανίχνευσης και αναγνώρισης αντικειμένων, μπορούμε να διαπιστώσουμε πως έχουμε την δυνατότητα να

χρησιμοποιήσουμε ένα κατηγοριοποιητή εικόνων για να αναγνωρίσουμε και να ανιχνεύσουμε αντικείμενα, στην παρούσα διπλωματική εργασία - οχήματα. Υπάρχουν δύο τρόποι με τους οποίους μπορεί να γίνει η αναγνώριση και η ανίχνευση αντικειμένων στους τομείς της μηχανικής μάθησης και όρασης και ψηφιακής επεξεργασίας εικόνας. Ο πρώτος τρόπος είναι με την χρήση κλασικών μεθόδων της μηχανικής όρασης και μάθησης για ανίχνευση αντικειμένων. Αυτές περιλαμβάνουν μεθόδους όπως τα κινητά παράθυρα στον χρόνο και οι πυραμίδες εικόνων (οι μέθοδοι που χρησιμοποιούνται στους HOG και linear SVM ανιχνευτές αντικειμένων). Ο δεύτερος τρόπος είναι με την χρήση προ εκπαιδευμένων δικτύων ως βασικό δίκτυο σε ένα ευρύτερο αρχιτεκτονικό πλαίσιο βαθιάς μάθησης και ανίχνευσης αντικειμένων (για παράδειγμα Fast R-CNN, SSD, YOLO) [9, 11].

Αξίζει να σημειωθεί πως ο δεύτερος τρόπος ανίχνευσης και αναγνώρισης αντικειμένων χρησιμοποιείται από εταιρίες κολοσσούς όπως το Facebook σε στόχο την αναγνώριση των προσώπων/ατόμων που βρίσκονται σε μια εικόνα που έχει αναρτηθεί στο γνωστό μέσο κοινωνικής δικτύωσης. Το γεγονός πως φέτος διεξάχθηκε το γνωστό #10yearchallenge σε πολλά μέσα κοινωνικής δικτύωσης δεν είναι τυχαίο καθώς υπάρχει ένα ανοικτό ερευνητικό πρόβλημα, το πρόβλημα της ανίχνευσης και αναγνώρισης προσώπου σε εικόνες και βίντεο. Κατά τη διάρκεια του #10yearchallenge οι χρήστες του προαναφερθέντος κοινωνικού δικτύου καλούνταν να ανεβάσουν μια φωτογραφία του εαυτού τους δίπλα από μια φετινή. Αυτή η ενέργεια θα βοηθούσε τα δίκτυα του Facebook να εκπαιδευτούν με περισσότερα δεδομένα εισόδου. Επομένως, η εκπαίδευση των δικτύων με εκατομμύρια εικόνες προσώπων στις οποίες παρουσιάζονταν τα ίδια πρόσωπα σε διαφορετικές ηλικίες, οδηγεί στην ακριβέστερη αναγνώριση ενός συγκεκριμένου προσώπου σε εικόνες ακόμη και αν το ίδιο πρόσωπο απεικονίζεται σε μεγαλύτερη ή μικρότερη ηλικία. Μια τέτοια εφαρμογή είναι χρήσιμη για το εν λόγω κοινωνικό δίκτυο ώστε να μπορεί να επισημαίνει με μεγαλύτερη ακρίβεια τις εικόνες των χρηστών του καθώς αυτοί μεγαλώνουν, λαμβάνοντας υπόψη και τις προηγούμενες τους φωτογραφίες που λήφθηκαν σε μικρότερη ηλικία. Η βελτίωση της αυτόματης αναγνώρισης προσώπου θα ανοίξει τον δρόμο για ασφαλέστερους τρόπους πρόσβασης σε κινητές συσκευές και

στα μέσα κοινωνικής δικτύωσης. Ακόμη, θα αναπτυχθούν πιο ακριβείς σαρωτές και αναγνωριστές προσώπου για ταυτοποίηση επιβατών στα μέσα μαζικής μεταφοράς (αεροδρόμια, λεωφορεία, μετρό)[18].

Το Facebook χρησιμοποιεί τον δεύτερο τρόπο ανίχνευσης και αναγνώρισης κατά την εργασία της αναγνώρισης προσώπων (ετικετοποίηση) σε φωτογραφίες. Οι αρχιτεκτονικές δικτύων που χρησιμοποιεί σε αυτή την εργασία και σε άλλες εργασίες μηχανικής όρασης είναι η AlexNet από το 2012-2014, η Fast-RCNN από το 2015-2016 και η MaskRCNN από το 2017 μέχρι σήμερα[18].

Ακόμη, το δημοφιλές μέσο κοινωνικής δικτύωσης χρησιμοποιεί ένα νέο αλγόριθμο του «κοντινότερου γείτονα» για να εντοπίσει ψευδείς ή spam λογαριασμούς, βίαιο ή μισαλλόδοξο (ρατσιστικό) περιεχόμενο. Επιπλέον, με την χρήση ενός μοντέλου κατηγοριοποίησης που μπορεί να αναγνωρίσει μέχρι και 10 000 διαφορετικές πράξεις σε ένα βίντεο, το σκανάρει καθώς αυτό ανεβαίνει στην πλατφόρμα κοινωνικής δικτύωσης. Συγκεκριμένα, το Facebook έχει δηλώσει πως σε ένα σύνολο δεδομένων που περιέχει 300000 βίντεο και 400 πράξεις η στοίβα της μηχανικής όρασης του μπορεί να κατηγοριοποιήσει ορθά ένα βίντεο με 82.8% ακρίβεια και με μειωμένο σφάλμα σε σχέση με προηγούμενα μοντέλα[18].

Η πλατφόρμα Facebook οδεύει προς την εκπαίδευση ενός είδους μάθησης που ονομάζεται αυτοεπιβλεπόμενη μάθηση. Σε αυτό το είδος μάθησης μη ετικετοποιημένα δεδομένα χρησιμοποιούνται μαζί με λίγα ετικετοποιημένα δεδομένα με στόχο την βελτίωση του ποσοστού ακριβείας κατηγοριοποίησης κατά την εκπαίδευση[18].



Σχήμα 3.17 Αρχιτεκτονικές Συνελικτικών Δικτύων που χρησιμοποιεί το Facebook στην εργασία της αναγνώρισης (ετικετοποίησης) προσώπων σε φωτογραφίες.

Κεφάλαιο 4

Βάσεις Δεδομένων και Επεξεργασία Δεδομένων

4.1 Βάσεις Δεδομένων με Εικόνες Οχημάτων

4.1.1 Βάση Δεδομένων Cal Tech Institute

4.1.2 Βάση Δεδομένων CBCL StreetScenes Challenge Framework

4.1.3 Βάση Δεδομένων UIUC

4.1.4 Σύνολο Οχημάτων DLR Munich

4.1.5 Σύνολο Δεδομένων Drone Stanford

4.1.6 Συλλογή VEDAI – Συλλογή Ανίχνευσης Οχημάτων σε Εναέριας Εικόνες

4.2 The Drone Dataset

4.2.1 Μορφή Βάσης Δεδομένων

4.2.2 Δεδομένα Εκπαίδευσης και Δεδομένα Επαλήθευσης

4.2.2.1 Μορφή Συνόλων Δεδομένων Εισόδου (Δεδομένων Εκπαίδευσης και Επαλήθευσης)

4.3 Μορφή Δεδομένων Εξόδου

4.3.1 Μορφή Δεδομένων Εξόδου Κατηγοριοποιητών Οχημάτων

4.3.2 Μορφή Δεδομένων Εξόδου Ανιχνευτών Οχημάτων

4.4 Μετρικές

4.5 Data Preprocessing

4.5.1 Aspect-Aware Preprocessor

4.5.2 Image to Array Preprocessor

4.5.3 Data Augmentation

3

4.6 Sliding Window

4.7 Image Pyramid

4.8 Non Maxima Suppression

4.9 Batch Processing

4.10 Ensembles – Dropout

4.1 Βάσεις Δεδομένων με Εικόνες Οχημάτων

Μέσα από την μελέτη της βιβλιογραφίας για την εκπόνηση της παρούσας διπλωματικής εργασίας, κρίθηκε αναγκαία η χρήση μιας βάσης δεδομένων με αρνητικές και θετικές εικόνες/παραδείγματα σύμφωνα με τα οποία θα εκπαιδευτούν και θα ελεγχθούν οι εφαρμογές (κατηγοριοποιητές και ανιχνευτές οχημάτων) που κατασκευάσαμε. Από την έρευνα που διεξάχθηκε στο διαδίκτυο για εντοπισμό βάσεων δεδομένων που αφορούν την ανίχνευση αντικειμένων βρέθηκαν τρεις βασικές κατηγορίες. Η πρώτη κατηγορία βάσεων δεδομένων σχετίζεται με την ανίχνευση προσώπου, η δεύτερη κατηγορία με την ανίχνευση πεζών και η τρίτη με την ανίχνευση οχημάτων. Στις επόμενες υποενότητες θα παρουσιάσουμε μερικές από τις διαδικτυακές βάσεις δεδομένων που εντοπίστηκαν και αφορούν το την ανίχνευση οχημάτων [9].

4.1.1 Βάση Δεδομένων Cal Tech Institute

Η βάση δεδομένων περιέχει 526 εικόνες οχημάτων σε μορφή jpeg. Η ανάλυση των εικόνων είναι 320x240 pixels. Στις φωτογραφίες παρουσιάζεται το πίσω μέρος των οχημάτων όπως ακριβώς λήφθηκαν (χωρίς καμία προ επεξεργασία) από αυτοκινητόδρομους της Νότιας Καλιφόρνιας [9].



Σχήμα 4.1. Βάση Δεδομένων Cal Tech Institute

4.1.2 Βάση Δεδομένων CBCL StreetScenes Challenge Framework

Αυτή η βάση δεδομένων είναι μια συλλογή εικόνων που αποτελείται από 3460 θετικά δείγματα και 87 αρνητικά, επισημάνσεις, λογισμικό και μετρικών απόδοσης με σκοπό την χρήση τους στην έρευνα της μηχανικής μάθησης και μηχανικής όρασης. Οι εικόνες της βάσης λήφθηκαν στην Βοστώνη και κάθε εικόνα επισημάνθηκε με ένα ορθογώνιο σχήμα που βρίσκεται γύρω από το αντικείμενο ενδιαφέροντος (π.χ. οχήματα, πεζοί, ποδήλατα, πεζοδρόμια, δέντρα, δρόμοι) [9].



Σχήμα 4.2. Βάση Δεδομένων CBCL StreetScenes Challenge Framework

4.1.3 Βάση Δεδομένων UIUC

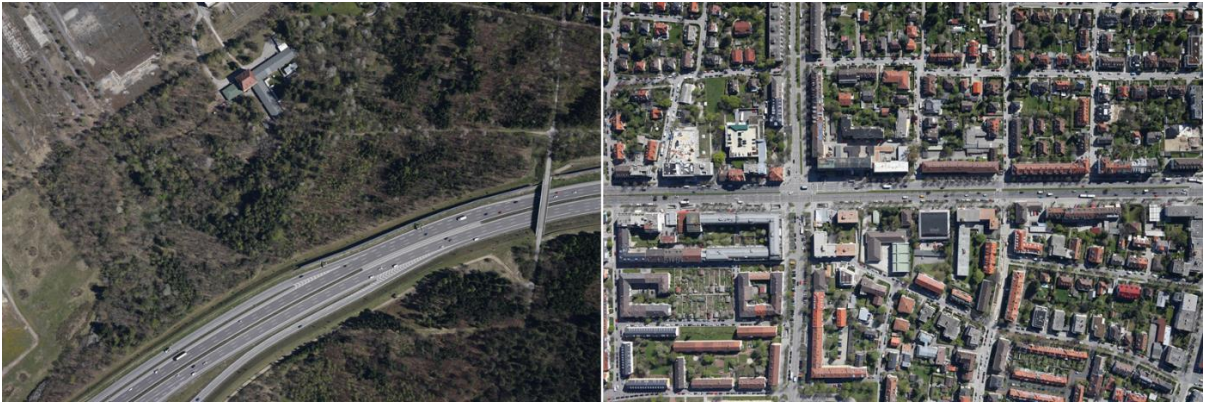
Περιέχει φωτογραφίες οχημάτων και συγκεκριμένα 550 θετικές φωτογραφίες και 500 αρνητικές. Οι λήψεις των φωτογραφιών είναι πλάγιες. Η ανάλυση τους είναι 100x40 pixels και βρίσκονται σε μορφή png. Ακόμη, η βάση δεδομένων διαθέτει ένα σύνολο 170 φωτογραφιών που μπορεί να χρησιμοποιηθεί ως σύνολο ελέγχου/επαλήθευσης [9].



Σχήμα 4.3. Βάση Δεδομένων UIUC

4.1.4 Σύνολο Οχημάτων DLR Munich

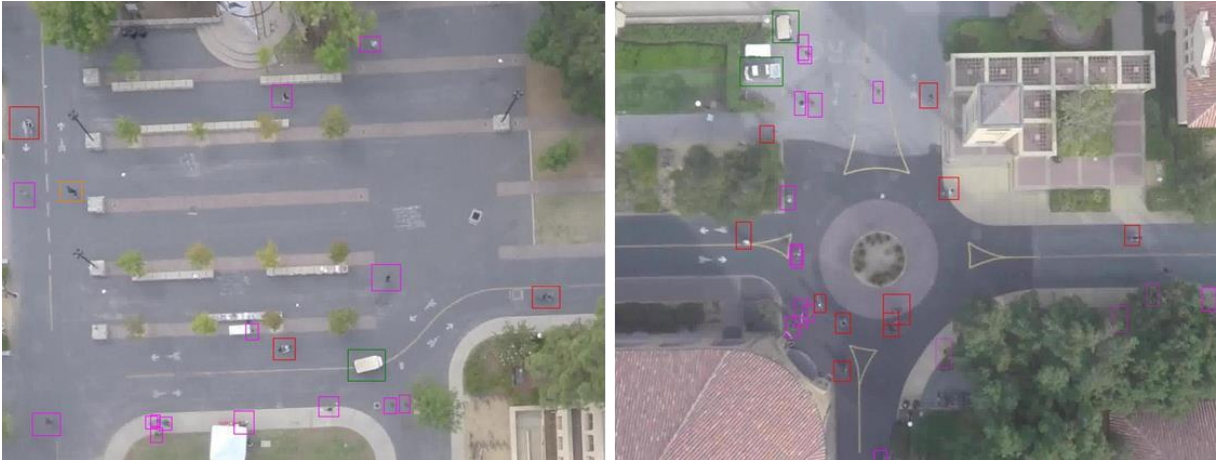
Το σύνολο οχημάτων αποτελείται από φωτογραφίες που ληφθηκαν από ύψος 1 Km με φωτογραφική κάμερα εστιακού βάθους 50mm και ανάλυσης 5616x3744 pixels. Επιπρόσθετα, οι φωτογραφίες αυτές έχουν αποθηκευτεί σε μορφή jpeg [9].



Σχήμα 4.4. Σύνολο Οχημάτων DLR Munich

4.1.5 Σύνολο Δεδομένων Drone Stanford

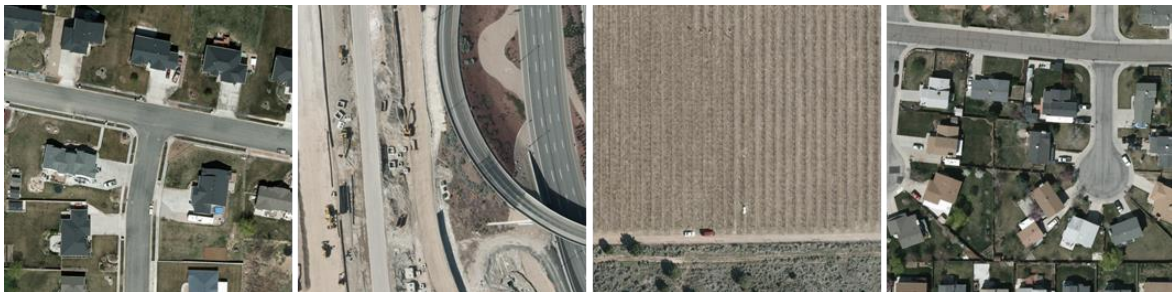
Η βάση δεδομένων που δημιούργησε το Stanford περιλαμβάνει εναέριας λήψεις που έγιναν στον χώρο της πανεπιστημιούπολης του εν λόγω πανεπιστημίου με την χρήση εναέριου μη επανδρωμένου μέσου (drone). Η συγκεκριμένη βάση δεδομένων περιέχει βίντεο και φωτογραφίες στις οποίες έγινε η επισήμανση σε αρκετά αντικείμενα που υπήρχαν μέσα σε αυτές όπως πεζοί, ποδηλάτες και διαφόρων ειδών οχήματα [9].



Σχήμα 4.5. Σύνολο Δεδομένων Drone Stanford. Τα οχήματα έχουν επισημανθεί με ορθογώνιο σχήμα πράσινου χρώματος (http://cvgl.stanford.edu/projects/uav_data)

4.1.6 Συλλογή VEDAI – Συλλογή Ανίχνευσης Οχημάτων σε Εναέριες Εικόνες

Η συγκεκριμένη συλλογή εικόνων περιέχει εναέριες λήψεις και δημιουργήθηκε για ανίχνευση πολύ μικρών οχημάτων. Οι λήψεις διεξάχθηκαν σε διαφορετικούς τύπους περιοχών π.χ αστικό πόλεις, δάση, προαστιακές και αγροτικές περιοχές. Οι εικόνες βρίσκονται σε μορφή jpeg και η ανάλυσή τους είναι 512x512 pixels και 1024x1024 pixels [9].



Σχήμα 4.6. Συλλογή VEDAI – Συλλογή Ανίχνευσης Οχημάτων σε Εναέριες Εικόνες

4.2 The Drone Dataset

Σε αυτό το τμήμα της διπλωματικής εργασίας θα παρουσιάσουμε το σύνολο δεδομένων που χρησιμοποιήθηκε για την εκπαίδευση, τον έλεγχο και την επαλήθευση των κατηγοριοποιητών εικόνων αλλά και για την εκπαίδευση και τον έλεγχο των ανιχνευτών των αντικειμένων [9].

Το σύνολο δεδομένων που θα παρουσιαστεί δημιουργήθηκε κατά την εκπόνηση μιας μεταπτυχιακής εργασίας με αντικείμενο την ανίχνευση και ιχνηλάτηση οχημάτων, που εντοπίστηκαν σε εναέριες λήψεις μη επανδρωμένων πτητικών μέσων, με κλασσικές μεθόδους μηχανικής μάθησης και μηχανικής όρασης (HOG & Linear SVM classifiers) [9].

Εξετάσαμε τα σύνολα δεδομένων ώστε να αποφασίσουμε κατά πόσο θα μπορούσαμε να χρησιμοποιήσουμε μερικά από εκείνα για την εκπόνηση της παρούσας διπλωματικής εργασίας. Εφόσον, το θέμα της παρούσας διπλωματικής εργασίας είναι η αναγνώριση και ανίχνευση οχημάτων σε εικόνες που λήφθηκαν από μη επανδρωμένα εναέρια πτητικά μέσα με την χρήση βαθιών συνελκτικών δικτύων, οι βάσεις δεδομένων ανάμεσα στις οποίες καλούμαστε να επιλέξουμε χρειάζεται να απαρτίζονται μόνο από εικόνες που προέρχονται από εναέριες λήψεις [9]. Επομένως, οι βάσεις δεδομένων που αποκλείσαμε αμέσως ήταν αυτές του Cal Tech Institute, του CBCL StreetScenes Challenge Framework και του UIUC συνόλου εικόνων. Κρίθηκε αναγκαία η χρήση της βάσης δεδομένων που κατασκευάστηκε για την ιχνηλάτηση και την αναγνώριση οχημάτων σε εναέριες λήψεις φωτογραφιών διότι η παρούσα διπλωματική εργασία ανήκει στο 2^ο στάδιο της ιχνηλάτησης οχημάτων εντός μιας συγκεκριμένης περιοχής περιορισμένου εύρους και γι' αυτό ο φακός χρειάζεται να εστιάσει σε κοντινό πλάνο σε συγκεκριμένες περιοχές ενδιαφέροντος (δρόμους με αυτοκίνητα ή άλλες εικόνες που δεν περιέχουν αυτοκίνητα).

4.2.1 Μορφή Βάσης Δεδομένων

Οι εναέρειες λήψεις έγιναν με το UAV Phantom 3 της DJI. Η κάμερα του εν λόγω μη επανδρωμένου εναέριου μέσου έχει την ικανότητα να καταγράψει βίντεο ανάλυσης ως και 4K. Η βάση δεδομένων που χρησιμοποιήθηκε ως σύνολο δεδομένων εισόδου για αυτή την εφαρμογή ονομάζεται Vertical Recorder Vehicles with UAS. Οι λήψεις για την δημιουργία της συγκεκριμένης βάσης δεδομένων πραγματοποιήθηκαν σε περιοχή της Ελλάδας (Νομός Μαγνησίας) και έλαβαν χώρα εντός του αστικού ιστού καθώς και στα περίχωρα της πόλης του Βόλου. Επιδιώχθηκε η συλλογή αρνητικών δειγμάτων από διαφορετικούς τύπους εδαφών ως προς την περιοχή (αστική/αγροτική), το χρωματισμό, την υφή και τις συνθήκες φωτισμού [9].

Το σύνολο δεδομένων εκπαίδευσης αποτελείται από εικόνες που λήφθηκαν από δύο περιοχές ενώ το σύνολο δεδομένων ελέγχου αποτελείται από εικόνες που λήφθηκαν από 4 διαφορετικές περιοχές για να συμπεριληφθεί μεγαλύτερη ποικιλία όσον αφορά τον περιβάλλοντα χώρο. Επομένως, η προαναφερθείσα βάση δεδομένων περιλαμβάνει λήψεις από αστικό, ημιαστικό και αγροτικό περιβάλλον. Η συγκεκριμένη βάση δεδομένων κυρίως περιλαμβάνει αυτοκίνητα ενώ τα υπόλοιπα οχήματα είναι λεωφορεία και φορτηγά. Η ανάλυση των εικόνων είναι 1920x1080 (εικόνα υψηλής ευκρίνειας), περιέχει πολλά pixels για την απεικόνιση των οχημάτων και παρέχει την δυνατότητα υποβιβασμού της ανάλυσης της εικόνας (διότι το εύρος ζώνης του παρόχου κινητής τηλεφωνίας δεν επαρκεί ή επειδή ο χρόνος σάρωσης της εικόνας από τον αλγόριθμο ανίχνευσης υπερβαίνει τον μέγιστο δυνατό χρόνο, για την διατήρηση της αναγνώρισης και της ιχνηλάτησης σε σχεδόν πραγματικό χρόνο. Η ευκρίνεια των οχημάτων είναι 85x35 pixels για μια εικόνα διαστάσεων 1920x1080, επιτρέπει την εύκολη οπτική αναγνώριση του τύπου των οχημάτων [9].

Δείγματα που ανήκουν στην κατηγορία/κλάση car της Βάσης Δεδομένων «The Drone Dataset»:



Σχήμα 4.7 car_451.jpeg



Σχήμα 4.10 car_41.jpeg

Σχήμα 4.8 car_159.jpeg



Σχήμα 4.11 car_19.jpeg

Σχήμα 4.9 car_138.jpeg

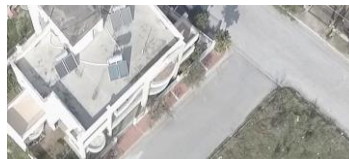


Σχήμα 4.12 car_63.jpeg

Δείγματα που ανήκουν στην κατηγορία/κλάση no-car της Βάσης Δεδομένων «The Drone Dataset»



Σχήμα 4.13 no-car_8.jpeg



Σχήμα 4.14 no-car_196.jpeg



Σχήμα 4.15 no-car_375.jpeg



Σχήμα 4.16 no-car_158.jpeg



Σχήμα 4.17 no-car_255.jpeg



Σχήμα.18 no-car_753.jpeg

4.2.2 Δεδομένα Εκπαίδευσης και Δεδομένα Επαλήθευσης

Ένα νευρωνικό δίκτυο για να έχει την δυνατότητα να προβλέψει μια απάντηση ή να κατηγοριοποιήσει ένα αντικείμενο χρειάζεται να εκπαιδευτεί με την χρήση ενός αλγορίθμου μηχανικής μάθησης και ενός συνόλου δεδομένων των δεδομένων εκπαίδευσης – στόχος των οποίων είναι να εκπαιδεύσουν το μοντέλο ώστε να αναγνωρίζει τα μοτίβα που περιέχουν και να ταυτοποιεί κάθε δεδομένο εκπαίδευσης σε μια κλάση δεδομένων. Αυτό, επιτυγχάνεται με ένα αλγόριθμο μάθησης όπως αυτούς που συζητήθηκαν στο . Η αξιολόγηση του μοντέλου, δηλαδή να μπορεί να γενικεύσει σύμφωνα με τα μοτίβα που έμαθε και όχι να τα παπαγαλίζει, γίνεται με την χρήση ενός συνόλου δεδομένων επαλήθευσης/ελέγχου που περιέχει εντελώς διαφορετικά παραδείγματα από το σύνολο δεδομένων εκπαίδευσης. Κατά την αξιολόγηση ενός μοντέλου μετρούμε πόσο

καλά το δίκτυο μας κατηγοριοποιήσει νέα παραδείγματα τα οποία δεν έχει ξαναδεί.

Γενικά, ένας τρόπος να χωριστεί το σύνολο δεδομένων σε σύνολο δεδομένων εκπαίδευσης και σύνολο δεδομένων επαλήθευσης είναι με τον κανόνα 80-20, 75-25, 90-10. Επιλέξαμε τον κανόνα 75-25 για να χωρίσουμε το σύνολο δεδομένων μας στο πρόβλημα αναγνώρισης και ανίχνευσης οχημάτων από εικόνες που λήφθηκαν από μη επανδρωμένα εναέρια μέσα. Σύμφωνα με τον κανόνα 75-25, το 75% των δεδομένων θα χρησιμοποιηθεί ως σύνολο δεδομένων εκπαίδευσης ενώ το υπόλοιπο 25% να χρησιμοποιηθεί ως σύνολο δεδομένων ελέγχου/επαλήθευσης, οι αναλογίες των συνόλων δεδομένων εκπαίδευσης και επαλήθευσης που θα χρησιμοποιηθούν εξαρτώνται από το πρόβλημα διαφορετικές αναλογίες μπορούν να παράξουν καλύτερα αποτελέσματα [12, 13].

4.2.2.1 Μορφή Συνόλων Δεδομένων Εισόδου (Δεδομένων Εκπαίδευσης και Επαλήθευσης)

Στους αλγορίθμους μηχανικής μάθησης και μηχανικής όρασης, η επίδοση του μοντέλου πρόβλεψης εξαρτάται από την ποιότητα των δεδομένων που δόθηκαν στον μοντέλο κατά την εκπαίδευση και επαλήθευση του. Επομένως, η επιτυχία ενός μοντέλου μηχανικής μάθησης συνδέεται με ένα καλά προετοιμασμένο σύνολο δεδομένων. Για να διασφαλιστεί η ποιότητα του συνόλου δεδομένων, αυτό πέρασε από μια διαδικασία επιλογής και καθαρισμού δεδομένων. Στην περίπτωση μας όπως εξηγήσαμε και σε προηγούμενες παραγράφους επιλέξαμε το σύνολο δεδομένων σύμφωνα με συγκεκριμένα κριτήρια μέσα από πολλές υποψήφιες βάσεις δεδομένων και έχει περάσει συγκεκριμένη προεπεξεργασία όπως καθαρισμό των εικόνων από θόρυβο. Την προεπεξεργασία των δεδομένων θα την συζητήσουμε στις υποενότητες που ακολουθούν [9].

Το σύνολο δεδομένων που χρησιμοποιήθηκε σε αυτή τη διπλωματική εργασία για σκοπούς κατηγοριοποίησης και επαλήθευσης της ανίχνευσης και αναγνώρισης οχημάτων σε

εικόνες που λήφθηκαν από μη επανδρωμένα εναέρια μέσα περιέχει συνολικά 1500 εικόνες που προέρχονται από την προαναφερθείσα βάση δεδομένων. Το σύνολο δεδομένων περιέχει 800 αρνητικά παραδείγματα, λήψεις που απεικονίζουν άδειους δρόμους ή διάφορες άλλες εικόνες όπως τοπία, κλπ και 700 θετικά παραδείγματα, λήψεις που απεικονίζουν δρόμους στους οποίους κινούνται διαφορετικοί τύποι οχημάτων. Οι εικόνες του συνόλου δεδομένων μας βρίσκονται σε μορφή jpeg και η ανάλυση τους είναι 96x96 dpi. Η ευκρίνεια των οχημάτων που ανήκουν στην κλάση θετικών παραδειγμάτων είναι 85x35 όπως αναφέρθηκε και σε προηγούμενη παράγραφο. Η ευκρίνεια των εικόνων που απεικονίζουν άδειους δρόμους όπως έχει προαναφερθεί είναι 1920x1080 ενώ η ευκρίνεια των τοπίων και των λοιπών εικόνων ποικίλει [9].

4.5 Data Preprocessing

Στην υποενότητα αυτή, θα εξηγήσουμε τις βασικές τεχνικές που χρησιμοποιήσαμε κατά την προεπεξεργασία των δεδομένων εισόδου στις εφαρμογές εξαγωγής χαρακτηριστικών οχημάτων και στους κατηγοριοποιητές οχημάτων. Η προεπεξεργασία δεδομένων, είναι μια διαδικασία που γίνεται πριν την εξαγωγή χαρακτηριστικών και την κατηγοριοποίηση και έχει ως στόχο την αύξηση του ποσοστού ακριβείας επαλήθευσης και την μείωση του ποσοστού σφάλματος επαλήθευσης [12, 13].

4.5.1 Simple Preprocessor and Image to Array Preprocessor – Εκπαίδευση και Επαλήθευση Κατηγοριοποιητών

Ο προεπεξεργαστής SimplePreprocessor() μας βοηθά να μεταβάλουμε τις διαστάσεις των εικόνων του συνόλου δεδομένων, μεταβάλλοντας τις διαστάσεις τους. Εφαρμόζουμε τους προεπεξεργαστές δεδομένων κατά την κατηγοριοποίηση οχημάτων [12,13]. .

Αρχικά, αρχικοποιούμε τον προεπεξεργαστή SimplePreprocessor(), δίνοντας του τις διαστάσεις που θέλουμε να αποκτήσουν τα δεδομένα π.χ SimplePreprocessor(32,32) (32x32 pixels input data size). Στη συνέχεια, αρχικοποιούμε τον προεπεξεργαστή ImageToArrayPreprocessor(). Φορτώνουμε το σύνολο δεδομένων, drone dataset, από τον δίσκο και μεταβάλλουμε την κλίμακα των pixels των εικόνων στο εύρος τιμών [0,1]. Ακολούθως, εφαρμόζουμε τον ImageToArrayPreprocessor για να ταξινομήσουμε την σειρά των καναλιών της εικόνας, αφού πρώτα μεταβάλουμε την κλίμακά της [12,13].

Επαναλαμβάνουμε την πιο πάνω διαδικασία για κάθε εικόνα του drone dataset. Επαναλαμβάνουμε ολόκληρη την διαδικασία για κάθε διαφορετική αρχιτεκτονική δικτύου με διαφορετικό μέγεθος εισόδου που θα εκπαιδεύσουμε και θα επαληθεύσουμε.

4.5.2 Aspect-Aware Preprocessor and Image to Array Preprocessor – Εκπαίδευση και Επαλήθευση fine-tuning/Transfer Learning Κατηγοριοποιητών

Ο Aspect-Aware Preprocessor διασφαλίζει πως διατηρείται το aspect ration τους, κατά την μεταβολή της κλίμακας τους. Εφαρμόζουμε τους συγκεκριμένους προεπεξεργαστές δεδομένων κατά την κατηγοριοποίηση οχημάτων με transfer learning [12,13].

Αρχικοποιούμε τον προεπεξεργαστή Aspect-Aware Preprocessor, δίνοντας του τις διαστάσεις που θέλουμε να αποκτήσουν τα δεδομένα π.χ AspectAwarePreprocessor (192,192) (192x192 pixels input data size). Στη συνέχεια, αρχικοποιούμε τον προεπεξεργαστή ImageToArrayPreprocessor(). Φορτώνουμε το σύνολο δεδομένων, drone dataset, από τον δίσκο και μεταβάλλουμε την κλίμακα των pixels των εικόνων στο εύρος τιμών [0,1]. Ακολούθως, εφαρμόζουμε τον ImageToArrayPreprocessor για να ταξινομήσουμε την σειρά των καναλιών της εικόνας, αφού πρώτα μεταβάλουμε την κλίμακά της [12, 13].

Επαναλαμβάνουμε την πιο πάνω διαδικασία για κάθε εικόνα του drone dataset. Επαναλαμβάνουμε ολόκληρη την διαδικασία για κάθε διαφορετική αρχιτεκτονική δικτύου με διαφορετικό μέγεθος εισόδου που θα εκπαιδεύσουμε και θα επαληθεύσουμε.

Image Preprocessing Pipeline:

1. Load an image from disk
2. Resize it to WxH pixels
3. Order the channel dimensions
4. Output the image

4.5.2 Data Augmentation

Η Κανονικοποίηση είναι μια τροποποίηση που εφαρμόζουμε στον αλγόριθμο εκπαίδευσης ώστε να μειώσουμε το σφάλμα της γενίκευσης του δικτύου αλλά όχι το σφάλμα εκπαίδευσής του. Με απλά λόγια, μια μέθοδος κανονικοποίησης προσπαθεί να μειώσει το σφάλμα επαλήθευσης με κόστος μια μικρή αύξηση του σφάλματος εκπαίδευσης.

Υπάρχουν διαφορετικά είδη κανονικοποίησης που είτε:

1. Τροποποίηση της Αρχιτεκτονικής του Δικτύου καθαυτή.
2. Επαύξηση των δεδομένων που δίνουμε στο Δίκτυο κατά την Εκπαίδευση.

Η τεχνική Dropout που εισήγαμε στην αρχιτεκτονική των δικτύων και θα εξηγήσουμε σε επόμενη υποενότητα είναι ένα παράδειγμα του πρώτου είδους κανονικοποίησης [12,13].

Η τεχνική Data Augmentation ανήκει στο δεύτερο είδος κανονικοποίησης. Η προαναφερθείσα τεχνική, αλλάζει την εμφάνιση των παραδειγμάτων εκπαίδευσης και τα δίνει στο δίκτυο για εκπαίδευση. Σκοπός της είναι να εκπαιδεύσει το δίκτυο, με καινούργια

δεδομένα εκπαίδευσης (που προέρχονται από τα πρωτότυπα δεδομένα εκπαίδευσης μας) ώστε αυτό να αποκτήσει καλύτερη γενίκευση και επομένως μας δίνει ψηλότερο ποσοστό ακριβείας επαλήθευσης και χαμηλότερο ποσοστό σφάλματος επαλήθευσης [12,13].

Μερικά από τα είδη του Data Augmentation που μπορούμε να εφαρμόσουμε στις πρωτότυπες εικόνες για να παράξουμε καινούργια δεδομένα εισόδου είναι [12,13]:

1. Translations
2. Περιστροφές
3. Αλλαγές Κλίμακας
4. Shearing
5. Horizontal Flips

Οι αλλαγές που προκύπτουν στην εμφάνιση των εικόνων, λόγω της εφαρμογής των προαναφερθέντων ειδών Data Augmentation σε αυτές, είναι αμελητέες. Επομένως δεν αλλοιώνεται το αποτέλεσμα του δικτύου με αυτήν [12,13].

Θα δούμε ακριβώς πως υλοποιήσαμε το Data Augmentation στο Κεφάλαιο 5.

4.6 Sliding window

Η μορφή ενός οχήματος σε μια συγκεκριμένη τοποθεσία μέσα στην εικόνα εξαρτάται από πολλά γειτονικά pixels τα οποία απαρτίζουν το όχημα. Αντί να εξετάζουμε με το δίκτυο κάθε pixel της εικόνας για να εντοπίσουμε το όχημα και την τοποθεσία του, δίνουμε σε αυτό πολλά pixels μαζί με την χρήση της τεχνικής sliding window. Το Sliding window είναι ένα παράθυρο με το οποίο δίνουμε στο δίκτυο έναν προκαθορισμένο αριθμό διαδοχικών pixels και το επιθυμητό αποτέλεσμα (ένα όχημα στην παρούσα εφαρμογή). Στην πρόβλεψη χρονοσειρών, το επιθυμητό αποτέλεσμα του παραθύρου είναι το επιθυμητό αποτέλεσμα του αμέσως επόμενου προτύπου εκπαίδευσης. Αυτό συμβαίνει διότι το

αποτέλεσμα σε μια συγκεκριμένη χρονική στιγμή στην πρόβλεψη προτύπων στον χρόνο εξαρτάται από το ιστορικό των εισόδων στις προηγούμενες χρονικές στιγμές [16].

Παρόλα αυτά, στην εφαρμογή ανίχνευσης οχημάτων σε εικόνες, η εύρεση ενός οχήματος εξαρτάται από τα γειτονικά pixels (προηγούμενα και επόμενα). Ως εκ τούτου το μεσαίο στοιχείο του παραθύρου είναι προβλέπεται και είναι το επιθυμητό αποτέλεσμα και αποτελεί επίσης το επιθυμητό αποτέλεσμα του παραθύρου. Σε κάθε επανάληψη, το παράθυρο μετακινείται κατά `StepSize`. Η παράμετρος αυτή, δηλώνει πόσα pixels θα αγνοήσουμε στην κατεύθυνση x και στην κατεύθυνση y . Δεν θα θέλαμε να ελέγχουμε κάθε pixel της εικόνας όσον αφορά το ερώτημα αν αυτό αποτελεί τμήμα ενός οχήματος (δηλ. `stepSize = 1`) διότι αυτό κοστίζει πολύ υπολογιστικά αφού εφαρμόζουμε τον κατηγοριοποιητή οχημάτων σε κάθε παράθυρο. Χρειάζεται να έχουμε υπόψη μας ότι όσο πιο μικρό είναι το μέγεθος των βημάτων (`stepSize`) που κάνει το παράθυρο τόσο περισσότερα παράθυρα θα πρέπει να εξετάσουμε. Συνήθως, η παράμετρος `stepSize` καθορίζεται βάσει του συνόλου δεδομένων που χρησιμοποιούμε για μια εφαρμογή και ρυθμίζεται έτσι ώστε να δίνει την βέλτιστη δυνατή επίδοση σε εικόνες του συνόλου δεδομένων. Πρακτικά, οι συνηθισμένες τιμές βημάτων παραθύρων που χρησιμοποιούνται είναι 4, 8 και 16 pixels. Ακόμη, το μέγεθος του παραθύρου ορίζεται από την παράμετρο `windowSize` και είναι το πλάτος και το μήκος του παραθύρου που ελέγχουμε όσον αφορά την ύπαρξη ενός οχήματος σε αυτό ή όχι. Το μέγεθος του παραθύρου (`windowSize`) μετριέται σε pixels [13, 15].

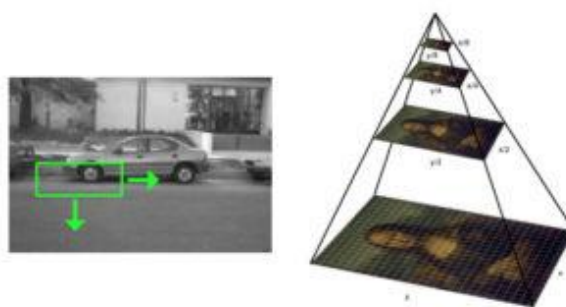
4.7 Image Pyramid

Μια πυραμίδα εικόνων είναι μια τεχνική που χρησιμοποιείται στο πεδίο της επεξεργασίας εικόνας. Η τεχνική αυτή, αναπαριστά μια εικόνα σε διαφορετικές κλίμακες. Επομένως, η χρήση αυτής της τεχνικής μας επιτρέπει να βρίσκουμε αντικείμενα που δεν έχουν το ίδιο μέγεθος και βρίσκονται μέσα στην ίδια εικόνα. Συνδυάζοντας αυτή την τεχνική με την

τεχνική του κινητού παραθύρου μπορούμε να βρούμε αντικείμενα με διαφορετική κλίμακα και σε διαφορετικές συντεταγμένες μέσα στην εικόνα. Ξεκινώντας από το πρώτο επίπεδο της πυραμίδας, ψάχνουμε σε ένα επίπεδο κάθε φορά το αντικείμενο που θέλουμε να εντοπίσουμε. Ακολούθως, ψάχνουμε σε κάθε επόμενο επίπεδο μέχρι να φτάσουμε στο τελευταίο επίπεδο της πυραμίδας [13, 14].

Στην βάση της πυραμίδα βρίσκεται η πρωτότυπη εικόνα στο πραγματικό της μέγεθος όσον αφορά το μήκος και το πλάτος της. Σε κάθε επόμενο επίπεδο η εικόνα λειαίνεται και οι διαστάσεις της μειώνονται μέχρι να φτάσει τις ελάχιστες δυνατές διαστάσεις της και να μην μπορεί να συμβεί άλλη υποδειγματοληψία . Συνήθως, η διαδικασία αυτή γίνεται με την χρήση Gaussian Blurring ενώ ταυτόχρονα διατηρείται το aspect ratio της εικόνας με κάθε υποδειγματολήψια [13, 14].

Μπορούμε να χρησιμοποιήσουμε την τεχνική αυτή στην βιβλιοθήκη OpenCV και να χρησιμοποιήσουμε την γλώσσα προγραμματισμού Python. Η πυραμίδα εικόνων χρειάζεται δύο παραμέτρους, την scale και την minSize. Η παράμετρος scale καθορίζει πόσο μειώνεται το μέγεθος της εικόνας σε κάθε επίπεδο της πυραμίδας. Η πυραμίδα αποτελείται από περισσότερα επίπεδα όσο πιο μικρή είναι η τιμή της παραμέτρου. Αντίθετα, η πυραμίδα εικόνας θα διαθέτει λιγότερα επίπεδα όσο πιο μεγάλη είναι η τιμή της παραμέτρου scale. Η παράμετρος minScale καθορίζει το ελάχιστο πλάτος και μήκος κάθε επιπέδου της πυραμίδας. Αν το μέγεθος μιας εικόνας γίνει μικρότερο από την τιμή της παραμέτρου minSize, τότε δεν κατασκευάζεται άλλο επίπεδο στην πυραμίδα [13,14].

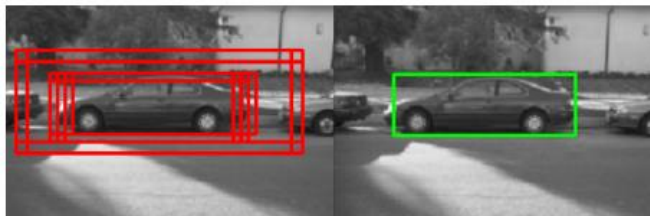


Σχήμα 4.19 – Sliding Window & Image Pyramid

4.8 Non Maxima Suppression

Ο συνδυασμός της χρήσης των τεχνικών του μετακινούμενου παραθύρου και της πυραμίδας εικόνων στην αναγνώριση και ανίχνευση οχημάτων, οδηγεί στην ανίχνευση του ίδιου αντικείμενου πολλές φορές (σχηματίζοντας πολλαπλά επικαλυπτόμενα κουτιά γύρω από αυτό). Όσο πιο πολλές φορές ανιχνεύεται ένα αντικείμενο τόσο μεγαλύτερη πιθανότητα αποκτά ως αντικείμενο μιας συγκεκριμένης κατηγορίας.

Η τεχνική Non-Maxima Suppression είναι ένα φίλτρο που αγνοεί τα πολλαπλά επικαλυπτόμενα κουτιά γύρω από τα αντικείμενα που θέλουμε να ανιχνεύσουμε. Συγκεκριμένα, δεν εμφανίζει τα κουτιά με χαμηλό σκορ εμπιστοσύνης (μικρότερη πιθανότητα) ενώ παρουσιάζει μόνο ένα κουτί γύρω από κάθε αντικείμενο ενδιαφέροντος (το κουτί με την μεγαλύτερη πιθανότητα να περιέχει το αντικείμενο που θέλουμε) [11,13]. Ένα κουτί με μεγάλη πιθανότητα να περιέχει το αντικείμενο ενδιαφέροντος είναι ένα κουτί που περιέχει μέσα το συγκεκριμένο αντικείμενο στο κέντρο (αποτελώντας το κύριο αντικείμενο στην συγκεκριμένη περιοχή ενδιαφέροντος).



Σχήμα 4.20 – Non Maxima Suppression

4.9 Batch Processing

Ο όρος batch processing αναφέρεται στην μαζική επεξεργασία δεδομένων που γίνεται από τα συνελκτικά δίκτυα. Τα προαναφερθέντα δίκτυα είναι πιο αποδοτικά όταν επεξεργάζονται τα δεδομένα ανά batch. Ένα συνελκτικό δίκτυο επεξεργάζεται ένα batch δεδομένων ανά μια εποχή εκπαίδευσης. Ένα συγκεκριμένο batch δεδομένων αποτελείται από ένα προκαθορισμένο πλήθος δεδομένων, παραδειγμάτων εκπαίδευσης ή επαλήθευσης και στα δεδομένα του συμβαίνει η ίδια μορφή επεξεργασίας (π.χ. κατηγοριοποίηση - batch κατηγοριοποίησης, εκπαίδευση - batch εκπαίδευσης δικτύου, επαλήθευση - batch επαλήθευσης δικτύου κλπ. Το μέγεθος ενός batch δηλαδή πόσα δεδομένα περιέχει είναι ένας αριθμός $2^{(n)}$ που εξαρτάται από την μνήμη τυχαίας προσπέλασης (RAM) του υπολογιστικού μας συστήματος (π.χ. 8GB RAM - $n = 8$). Τα συνηθισμένα μεγέθη batch size που χρησιμοποιούνται κατά την εκπαίδευση ή την επαλήθευση είναι batch size = 32 ενώ κατά την διεξαγωγή πειραμάτων batch size > 32 and batch size < 512 [13].

4.10 Ensembles – Dropout

Στην μηχανική μάθηση, η χρήση της τεχνικής κανονικοποίησης Ensembles, είναι ένας καλός τρόπος βελτίωσης της απόδοσης του μοντέλου μάθησης, αποφεύγοντας την υπερεκπαίδευση. Υπάρχουν πολλές τεχνικές ensembles. Στην παρούσα διπλωματική χρησιμοποιήσαμε την συνάρτηση Dropout. Η προαναφερθείσα συνάρτηση μας βοηθά να αποφύγουμε την υπερεκπαίδευση και να γλιτώσουμε αρκετό χρόνο εκπαίδευσης συνδυάζοντας τις προβλέψεις πολλών διαφορετικών δικτύων κατά την επαλήθευση. Η ιδέα είναι να εισάξουμε ένα νέο επίπεδο που να αφαιρεί τυχαία νευρώνες του δικτύου μαζί με τις συνδέσεις τους (**από το προηγούμενο επίπεδο στο επόμενο**), κατά την

εκπαίδευση. Αυτό αποτρέπει την προσαρμογή των νευρώνων και έτσι δεν γνωρίζουν εκ των προτέρων πως αναπαρίσταται μια συγκεκριμένη κατηγορία/κλάση δεδομένων.

Κατά την εκπαίδευση, λαμβάνουμε δείγματα dropout από ένα εκθετικό αριθμό διαφορετικών αραιών δικτύων. Κατά την επαλήθευση υπολογίζουμε τον μέσο όρο των προβλέψεων των αραιών δικτύων χρησιμοποιώντας ένα μη αραιό δίκτυο που έχει μικρότερα βάρη. Επομένως, αυτή η τεχνική μειώνει την υπερεκπαίδευση και δίνει περισσότερες βελτιώσεις σε σύγκριση με άλλες μεθόδους κανονικοποίησης. Αυτή η τεχνική βελτιώνει την επίδοση των νευρωνικών δικτύων σε επιβλεπόμενες εργασίες όπως η αναγνώριση ομιλίας, η αναγνώριση αντικειμένων, η κατηγοριοποίηση αρχείων και μας βοηθά να λάβουμε εξαιρετικά αποτελέσματα σε αρκετά σύνολα δεδομένων. Ως εκ τούτου, η προαναφερθείσα τεχνική χρησιμοποιείται στους διάφορους κατηγοριοποιητές οχημάτων που κατασκευάσαμε κατά την εκπαίδευση τους [13].

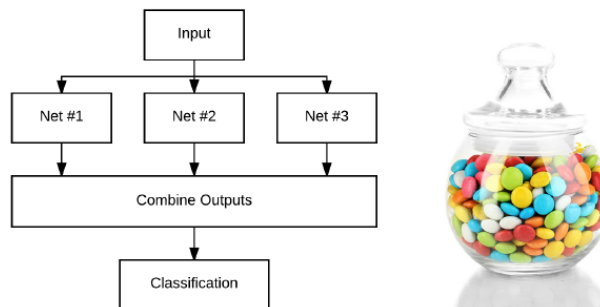


Figure 6.2: **Left:** An ensemble of neural networks consists of *multiple* networks. When classifying an input image the data point is passed to each network where it classifies the image *independently* of all other networks. The classifications across networks are then averaged to obtain the final prediction. **Right:** Ensemble methods are possible due to Jensen's Inequality. By averaging guesses as to the number of candies in the jar, we can better approximate the true number of candies.

Σχήμα 4.21 Ensembles [12,13]

*Υπερεκπαίδευση: Το δίκτυο να μάθει παπαγαλία τα δεδομένων εισόδου και να τα παρουσιάζει ως αποτελέσματα εξόδου.

Κεφάλαιο 5

Μεθοδολογία

5.1 Object Detection Pipeline

5.2 Μεθοδολογία Ατομικής Διπλωματικής Εργασίας

5.3 Χρήση Προεκπαιδευμένων Δικτύων Μάθησης

5.4 Λογισμικό που Χρησιμοποιήθηκε στην Υλοποίηση

5.4.1 OpenCV

5.4.2 TensorFlow και Keras

5.4.3 Scikit (Scikit Learn,.., etc)

5.4.4 NumPy

5.5 Transfer Learning

5.5.1 Transfer Learning, Fine-tuning και Εγχείρηση Συνελικτικού Δικτύου

5.6 Εξαγωγή Χαρακτηριστικών με την Χρήση Προ εκπαιδευμένων Δικτύων – Παράδειγμα VGG16

5.6.1 Εξαγωγή Χαρακτηριστικών Οχημάτων –VGG16

5.6.2 Εξαγωγή Χαρακτηριστικών Οχημάτων – ResNet50

5.6.3 Εξαγωγή Χαρακτηριστικών Οχημάτων – ShallowNet

5.6.4 Εξαγωγή Χαρακτηριστικών Οχημάτων – MobileNet

5.6.5 Εξαγωγή Χαρακτηριστικών Οχημάτων– VGG16 finetuned

5.6.6 Εξαγωγή Χαρακτηριστικών Οχημάτων – MobileNet finetuned

5.7 Δημιουργία Δειγμάτων Εκπαίδευσης και Ελέγχου για DCNNs

5.7.1 Δημιουργία Δειγμάτων Εκπαίδευσης και Ελέγχου για Προεκπαιδευμένη

Έκδοση Συνελικτικού Δικτύου - VGG16

5.7.2 Δημιουργία Δειγμάτων Εκπαίδευσης και Ελέγχου για Προεκπαιδευμένη

Έκδοση Συνελικτικού Δικτύου - ResNet50

5.7.3 Δημιουργία Δειγμάτων Εκπαίδευσης και Ελέγχου για Προεκπαιδευμένη

Έκδοση Συνελικτικού Δικτύου - ShallowNet

5.7.4 Δημιουργία Δειγμάτων Εκπαίδευσης και Ελέγχου για Προεκπαιδευμένη

Έκδοση Συνελικτικού Δικτύου - MobileNet

5.7.6 Δημιουργία Δειγμάτων Εκπαίδευσης και Ελέγχου για Προεκπαιδευμένη fine-tuned Έκδοση Συνελικτικού Δικτύου - VGG16 finetuned

5.7.7 Δημιουργία Δειγμάτων Εκπαίδευσης και Ελέγχου για Προεκπαιδευμένη fine-tuned Έκδοση Συνελικτικού Δικτύου - MobileNet finetuned

5.8 Εγχείρηση Συνελικτικών Δικτύων

5.8.1 Εγχείρηση Προεκπαιδευμένων Εκδόσεων Συνελικτών Δικτύων

5.8.1.1 Εγχείρηση Προεκπαιδευμένης Έκδοσης Συνελικτού Δικτύου - VGG16

5.8.1.2 Εγχείρηση Προεκπαιδευμένης Έκδοσης Συνελικτού Δικτύου - ResNet50

5.8.1.3 Εγχείρηση Προεκπαιδευμένης Έκδοσης Συνελικτού Δικτύου - ShallowNet

5.8.1.4 Εγχείρηση Προεκπαιδευμένης Έκδοσης Συνελικτού Δικτύου - MobileNet

5.8.2 Εγχείρηση Προεκπαιδευμένων Εκδόσεων fine-tuned Συνελικτών Δικτύων

5.8.2.1 Εγχείρηση Προεκπαιδευμένης Έκδοσης fine-tuned Συνελικτού Δικτύου - VGG16 fine-tuned

5.8.2.2 Εγχείρηση Προεκπαιδευμένης Έκδοσης fine-tuned Συνελικτού Δικτύου - MobileNet fine-tuned

5.10 Ανίχνευσης Οχημάτων με την χρήση Συνελικτικών Δικτύων και Υλοποίηση της

5.10.1 Υλοποίηση Sliding Window

5.10.2 Υλοποίηση Image Pyramid

5.10.3 Batch Processing

5.10.4 Non maxima Suppression

5.1 Object Detection Pipeline

Όταν κατασκευάζουμε έναν ανιχνευτή αντικειμένων βαθιάς μηχανικής μάθησης με την κλασικών μεθόδων ανίχνευσης αντικειμένων χρειαζόμαστε 4 συστατικά [11, 13]:

1.Sliding Window

Το sliding Window ελέγχει την εικόνα από αριστερά προς δεξιά και από πάνω προς τα κάτω, κατηγοριοποιώντας κάθε περιοχή ενδιαφέροντος (ROI) που βρίσκει μπροστά του. Το sliding window μας επιτρέπει να εντοπίζουμε που ακριβώς βρίσκεται ένα αντικείμενο, (στις εφαρμογές μας, ένα όχημα), μέσα στην εικόνα) [11, 13].

2.Image Pyramid

Μια πυραμίδα εικόνων μειώνει/ή αυξάνει διαδοχικά το μέγεθος της εικόνας εισόδου. Επίσης, ονομάστηκε πυραμίδα εικόνων διότι οι εικόνες τοποθετούνται σε επίπεδα σύμφωνα με το μέγεθος τους. Στην βάση της πυραμίδας βρίσκεται η εικόνα με την μεγαλύτερη κλίμακα ενώ στην κορυφή η ίδια εικόνα στην μικρότερη δυνατή κλίμακα. Το sliding window εξετάζει κάθε κλίμακα της πυραμίδας εικόνων, επιτρέποντας μας να ανιχνεύσουμε τα οχήματα που βρίσκονται σε πιο μικρή ή σε πιο μεγάλη κλίμακα δηλαδή πιο κοντά ή πιο μακριά από την κάμερα [11, 13].

3. Non-Maxima Suppression

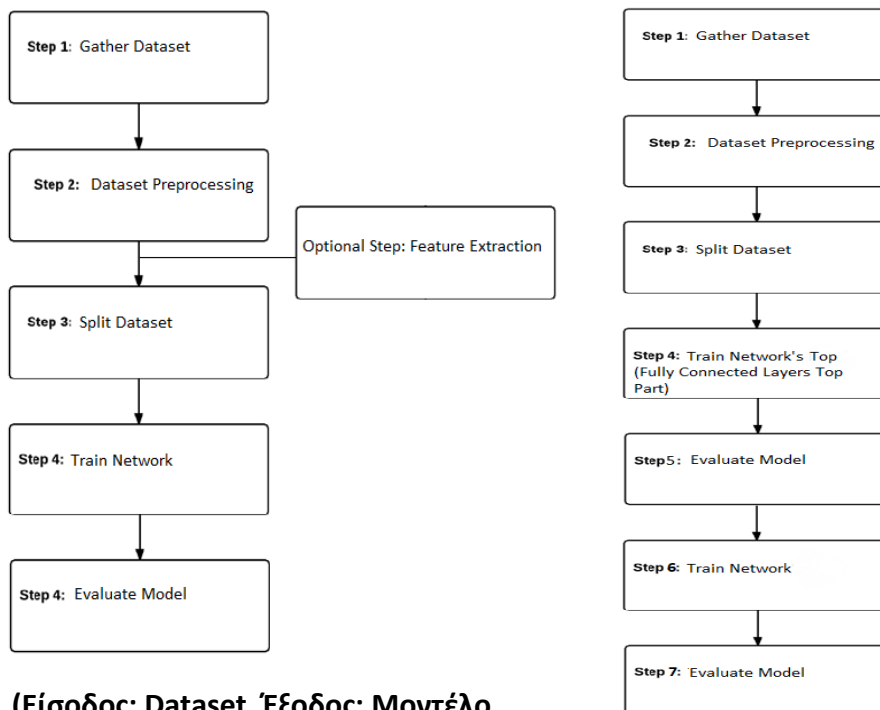
Τα sliding window και image pyramid μαζί ελέγχουν κάθε τοποθεσία της εικόνας σε κάθε κλίμακα και θα ανιχνεύει το ίδιο αντικείμενο πολλές φορές μέσα στην εικόνα, σχηματίζοντας πολλαπλά κουτιά γύρω του. Έτσι εφαρμόζουμε την Non-Maxima Suppression, για να κρατήσουμε την πρόβλεψη με την υψηλότερη εμπιστοσύνη.

Επομένως, μετά την εφαρμογή του non-maxima suppression, λαμβάνουμε τα τελικά κουτιά πρόβλεψης του ανιχνευτή [11,13].

4. Batch Processing

Τέλος, εφαρμόζουμε batch processing στις περιοχές ενδιαφέροντος (ROIs) για να διασφαλίσουμε πως ο ανιχνευτής οχημάτων εκτελείται όσο πιο γρήγορα γίνεται [13].

5.2 Μεθοδολογία Ατομικής Διπλωματικής Εργασίας



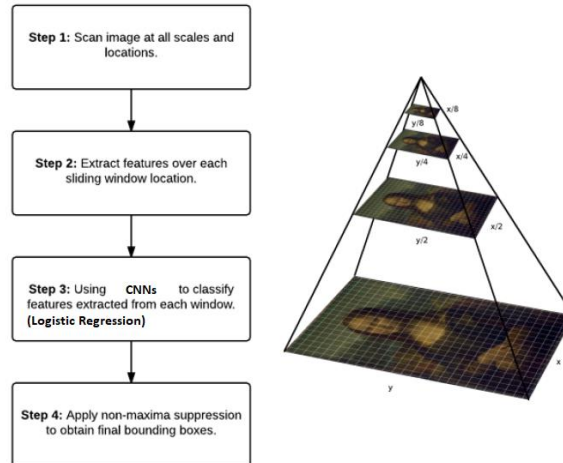
(Είσοδος: Dataset, Έξοδος: Μοντέλο Κατηγοριοποίησης Οχημάτων & Ποσοστά Εκπαίδευσης, Επαλήθευσης και Χρόνος Εκπαίδευσης). Ισχύει το ίδιο και στο Σχήμα 5.2.

Σχήμα 5.1 – Μεθοδολογία Μέρος Α΄:

Εξαγωγή Χαρακτηριστικών Οχημάτων και Δημιουργία Μοντέλων Κατηγοριοποίησης

Σχήμα 5.2 – Μεθοδολογία Μέρος Α΄:

Εξαγωγή Χαρακτηριστικών Οχημάτων και Δημιουργία Μοντέλων Κατηγοριοποίησης (Transfer Learning).



Σχήμα 5.3 – Μεθοδολογία Μέρος Β': Ανίχνευση Οχημάτων σε Εικόνες

5.3 Χρήση Προεκπαιδευμένων Δικτύων Μάθησης

Ένα προεκπαιδευμένο δίκτυο μάθησης είναι ένα δίκτυο που έχει ήδη εκπαιδευτεί με στόχο την επίλυση ενός συγκεκριμένου προβλήματος. Τα βάρη του δικτύου έχουν εκπαιδευτεί για να επιλύουν ένα πρόβλημα A. Μπορούμε να χρησιμοποιήσουμε μερικά από τα επίπεδα του δικτύου αυτού, που περιέχουν προεκπαιδευμένα βάρη, ώστε να εκπαιδεύσουμε το δίκτυο μας για να επιλύει ένα πρόβλημα B. Υπάρχουν αρκετές βιβλιοθήκες που περιέχουν προεκπαιδευμένα δίκτυα μάθησης. Μερικές από αυτές είναι η βιβλιοθήκη Keras, MicrosoftML R package και microsoftml Python package. Στην πρώτη και στην τρίτη βιβλιοθήκη πετυχαίνουμε την πρόσβαση μας μέσω της γλώσσας προγραμματισμού Python ενώ στην δεύτερη βιβλιοθήκη μέσω της γλώσσας προγραμματισμού R. Τρία δημοφιλή προεκπαιδευμένα δίκτυα είναι τα ResNet, AlexNet και Inception-v3. Αυτά τα δίκτυα έλαβαν μέρος στον ετήσιο διαγωνισμό ILSRC και πέτυχαν εξαιρετικά αποτελέσματα στην κατηγοριοποίηση εικόνων σε 1000 κατηγορίες [6].

5.4 Γλώσσα Προγραμματισμού και Βιβλιοθήκες Λογισμικού που Χρησιμοποιήθηκαν στην Υλοποίηση

Όσον αφορά την υλοποίηση της παρούσας διπλωματικής εργασίας αποφασίσαμε να χρησιμοποιήσουμε την γλώσσα προγραμματισμού Python μιας και είναι μια εύκολη γλώσσα στην εκμάθηση, διαθέτει διαισθητική σύνταξη και πάρα πολλές βιβλιοθήκες με αλγορίθμους και δίκτυα βαθιάς μάθησης.

Η κύρια βιβλιοθήκη βαθιάς μάθησης που θα χρησιμοποιήσουμε είναι το Keras. Η βιβλιοθήκη αυτή συντηρείται από τον Francois Chollet, ένα ερευνητή βαθιάς μάθησης και μηχανικό λογισμικού στην Google. Μπορούμε να χρησιμοποιήσουμε ως βοηθητική βιβλιοθήκη στο backend, την βιβλιοθήκη Theano ή την βιβλιοθήκη TensorFlow. Ακόμη, χρησιμοποιούμε μερικές βιβλιοθήκες μηχανικής όρασης, ψηφιακής επεξεργασίας εικόνας και μηχανικής μάθησης όπως οι OpenCV, scikit-learn, scikit-image, NumPy κλπ [12, 13].

5.4.1 OpenCV

Η βιβλιοθήκη OpenCV μας εξοπλίζει με τις κατάλληλες λειτουργίες της επεξεργασίας εικόνας όπως. Μερικές από αυτές είναι να φορτώσουμε μια εικόνα, να παρουσιάσουμε μια εικόνα στην οθόνη και άλλες βασικές λειτουργίες [12, 13].

5.4.2 TensorFlow και Keras

Είναι ένα γενικής χρήσης αφαιρετικό πρότυπο γράφων υπολογισμού. Αυτοί οι Γράφοι χρησιμοποιούνται τόσο στην βαθιά μηχανική μάθηση όσο και σε άλλες εφαρμογές που δεν έχουν καμία σχέση αυτήν. Το TensorFlow είναι το υπολογιστικό backend κομμάτι των εφαρμογών μας. Μπορεί κανείς να φανταστεί το TensorFlow ως μια μηχανή αυτοκινήτου. Έχεις την δυνατότητα να αντικαταστήσεις κομμάτια της μηχανής του αυτοκινήτου σου. Από την άλλη, το Keras είναι μια βιβλιοθήκη βαθιάς μηχανικής μάθησης, που σχεδιάστηκε με αυτό τον σκοπό, παρέχοντας καλοσχεδιασμένες διεπαφές προγραμματιστικές

διεπαφές εφαρμογών ώστε να καταστεί ευκολότερη η δημιουργία βαθιών νευρωνικών δικτύων. Κάτω από αυτήν την σκοπιά το Keras χρησιμοποιεί είτε την βιβλιοθήκη TensorFlow είτε την Theano ως υπολογιστικό backend, δίνοντας του την δυνατότητα να τις εκμεταλλευτεί ως υπολογιστικές μηχανές [12, 13].

5.4.3 Scikit (Scikit Learn,..., etc)

Η βιβλιοθήκη Scikit είναι μια βιβλιοθήκη ειδικά σχεδιασμένη για υλοποίηση εφαρμογών μηχανικής μάθησης. Περιέχει πολλούς αλγόριθμους και συναρτήσεις μηχανικής μάθησης. Χρησιμοποιούμε αυτή την βιβλιοθήκη κυρίως για τους αλγόριθμους μηχανικής μάθησης (Logistic Regression, SGD, κλπ.) που παρέχει [12, 13].

5.4.4 NumPy

Η βιβλιοθήκη NumPy είναι βασική βιβλιοθήκη της γλώσσας προγραμματισμού Python. Διαθέτει χρήσιμες μαθηματικές συναρτήσεις και βασικές συναρτήσεις πινάκων (numpy arrays) [12, 13].

5.5 Transfer Learning

Το Transfer Learning είναι η εφαρμογή της γνώσης που αποκτήθηκε κατά την εκπαίδευση ενός μοντέλου μηχανικής μάθησης για την επίλυση ενός προβλήματος σε άλλο διαφορετικό πρόβλημα που όμως σχετίζεται με το αρχικό πρόβλημα. Για παράδειγμα, αν εκπαιδεύσαμε ένα απλό κατηγοριοποιητή για να αναγνωρίζει αν μια εικόνα περιέχει ένα καπέλο, μπορούμε να χρησιμοποιήσουμε την γνώση που απόκτησε το μοντέλο κατά στην αναγνώριση άλλων αντικειμένων. Στην μεθοδολογία transfer learning εκμεταλλευόμαστε

την γνώση που αποκτήσαμε σε μια εργασία για να βελτιώσουμε την γενίκευση σε μια άλλη, μεταφέροντας τα βάρη του δικτύου που έμαθε να εκτελεί την εργασία A σε ένα άλλο νέο δίκτυο που θα μάθει να εκτελεί την εργασία B. Η ιδέα είναι να χρησιμοποιήσουμε την γνώση που απέκτησε ένα μοντέλο κατά την επίλυση μιας εργασίας με ένα σύνολο δεδομένων που περιελάμβανε πολλά ετικετοποιημένα δεδομένα εκπαίδευσης σε ένα νέο μοντέλο που θα μάθει να επιλύει μια άλλη σχετική εργασία στην οποία δεν υπάρχουν πολλά δεδομένα εκπαίδευσης [6].

Το transfer learning χρησιμοποιείται σε εφαρμογές επεξεργασίας φυσικής γλώσσας και μηχανικής όρασης. Η χρήση του transfer learning στις εφαρμογές φυσικής γλώσσας είναι απαραίτητη καθώς σε αυτές τα δίκτυα εκπαιδεύονται με γνώσεις εμπειρών επαγγελματιών και χρειάζεται να παρέχονται σε αυτά μεγάλα σύνολα ετικετοποιημένων δεδομένων. Επομένως, χρειαζόμαστε μερικές ώρες ή λίγα λεπτά για να εκπαιδεύσουμε τα βαθιά νευρωνικά δίκτυα σε πολύπλοκες εργασίες αντί να τα εκπαιδεύσουμε από την αρχή (ο χρόνος εκπαίδευσης ενός συνελκτικού δικτύου χωρίς προεκπαιδευμένα βάρη είναι 2-3 εβδομάδες ανά εκτέλεση) [6].

Η μεθοδολογία σχεδιασμού Transfer Learning χρησιμοποιείται σε εφαρμογές στα πεδία της Μηχανικής Όρασης, της Επεξεργασίας της Φυσικής Γλώσσας, όπως σε εργασίες Sentiment Analysis λόγω της μεγάλης υπολογιστικής δύναμης που απαιτείται σε αυτές [6].

Στο Transfer Learning προσπαθούμε να μεταφέρουμε όση περισσότερη γνώση μπορούμε από μια εργασία στην οποία εκπαιδεύτηκε κάποιο μοντέλο σε μια άλλη. Η γνώση μπορεί να έχει πολλές και διαφορετικές μορφές και εξαρτάται από το πρόβλημα που θέλουμε να επιλύσουμε και τα δεδομένα που θα χρησιμοποιήσουμε κατά την επίλυσή του [6].

5.5.1 Transfer Learning, Fine-tuning, Εγχείρηση Συνελκτικού Δικτύου και Προετοιμασία πριν την Εγχείρηση

Το fine tuning είναι ένα είδος transfer learning. Εφαρμόζουμε το fine-tuning/transfer learning σε μοντέλα βαθιάς μάθησης που έχουν ήδη εκπαιδευτεί σε ένα συγκεκριμένο σύνολο δεδομένων. Αυτά τα δίκτυα είναι αρχιτεκτονικές βαθιών συνελκτικών δικτύων όπως VGG, ResNet κλπ που έχουν εκπαιδευτεί στο σύνολο δεδομένων ImageNet. Τα δίκτυα αυτά είναι καλοί εξαγωγείς χαρακτηριστικών αφού περιέχουν φίλτρα που μπορούν να χρησιμοποιηθούν σε νέα σύνολα και νέες κλάσεις δεδομένων για τις οποίες το δίκτυο δεν έχει εκπαιδευτεί. Το fine-tuning είναι ουσιαστικά η τροποποίηση της αρχιτεκτονικής του δικτύου (η λεγόμενη εγχείρηση δικτύου).

Ο όρος εγχείρηση δικτύου αναφέρεται στην αφαίρεση των πρωτότυπων επιπέδων της αρχιτεκτονικής του δικτύου και η προσθήκη νέων. Ονομάστηκε έτσι επειδή μοιάζει με ιατρική εγχείρηση. Η εγχείρηση του δικτύου πραγματοποιείται με πολλή ακρίβεια και αφαιρούνται συστατικά του δικτύου που είναι προβληματικά και στην θέση τους προθέτονται νέα λειτουργικά συστατικά. Στην συνέχεια, κατά την εκπαίδευση ενός δικτύου, μεταβάλλονται τόσο τα βάρη των πρωτότυπων επιπέδων όσο και τα βάρη των νέων.

Για να καταλάβουμε πως λειτουργεί η διαδικασία fine-tuning/transfer learning, θα δούμε ένα παράδειγμα που περιλαμβάνει τα επίπεδα της αρχιτεκτονικής VGG16. Γνωρίζουμε πως το τελευταίο τμήμα του δικτύου αποτελείται από πλήρως συνδεδεμένα επίπεδα μαζί με softmax κατηγοριοποιητές. Κατά το fine-tuning/transfer learning, αφαιρούμε το τελευταίο πλήρως συνδεδεμένο τμήμα του δικτύου όπως ακριβώς κάνουμε και στην εξαγωγή χαρακτηριστικών. Όμως, σε αντίθεση με αυτήν, όταν πραγματοποιούμε το fine-tuning, δημιουργούμε ένα νέο τελευταίο πλήρως συνδεδεμένο τμήμα του δικτύου και το τοποθετούμε πάνω από τα επίπεδα της πρωτότυπης αρχιτεκτονικής. Θυμίζουμε πως δεν χρησιμοποιούμε καθόλου τα παλιά πλήρως συνδεδεμένα επίπεδα της πρωτότυπης αρχιτεκτονικής ενώ εκπαιδεύουμε τα νέα πλήρως συνδεδεμένα επίπεδα μαζί με την υπόλοιπη πρωτότυπη αρχιτεκτονική σύμφωνα με το σύνολο δεδομένων μας .

Στις πλείστες περιπτώσεις, το νέο πλήρες συνδεδεμένο τμήμα του δικτύου μας θα έχει λιγότερες παραμέτρους σε σχέση με το πρωτότυπο αλλά αυτό εξαρτάται πάντα από το σύνολο δεδομένων που διαθέτουμε. Το νέο πλήρες συνδεδεμένο τμήμα του δικτύου αρχικοποιείται τυχαία (όπως κάθε επίπεδο σε ένα καινούργιο δίκτυο και συνδέεται μαζί με το πρωτότυπο δίκτυο και είναι έτοιμο για εκπαίδευση. Τα επίπεδα συνέλιξης μας έχουν ήδη μάθει συγκεκριμένα φίλτρα διαχωρισμού ενώ τα πλήρως συνδεδεμένα επίπεδα είναι καινούργια και οι τιμές των βαρών τους είναι τυχαίες. Αν επιτρέψουμε στην κλίση να πάει προς τα πίσω (στο δίκτυο), από αυτές τις τυχαίες τιμές μέχρι την αρχή του δικτύου, θα καταστρέψουμε τα χαρακτηριστικά. Για να ξεπεράσουμε αυτό το εμπόδιο, παγώνουμε όλα τα επίπεδα του πρωτότυπου δικτύου αφήνοντας το τελευταίο πλήρως συνδεδεμένο του δικτύου να ζεσταθεί. Ξεκινώντας την διαδικασία του fine-tuning, παγώνουμε όλα τα επίπεδα συνέλιξης που υπάρχουν στο δίκτυο και αφήνουμε τη κλίση να μεταφερθεί προς τα πίσω μέσω των πλήρως συνδεδεμένων επιπέδων.

Το fine tuning/transfer learning είναι μια ισχυρή μέθοδος για να δημιουργήσουμε κατηγοριοποιητές εικόνων από προεκπαιδευμένα συνελκτικά δίκτυα που εκπαιδεύονται σε διαφορετικά σύνολα δεδομένων. Το κύριο αρνητικό της είναι πως χρειάζεται προσεχτική επιλογή των παραμέτρων του τελευταίου τμήματος του δικτύου διότι αυτές καθορίζουν σε μεγάλο βαθμό το ποσοστό ακριβείας του δικτύου. Ένα άλλο αρνητικό στοιχείο της μεθόδου αυτής είναι πως σε μικρά σύνολα δεδομένων το νέο τελευταίο πλήρως συνδεδεμένο τμήμα του δικτύου δυσκολεύεται να μάθει γι' αυτό παγώνουμε τα υπόλοιπα επίπεδα πριν να το εκπαιδεύσουμε και χρησιμοποιούμε διαφορετικούς αλγόριθμους εκπαίδευσης (SGD, RMSProp, Adam, Nadam, κλπ). Το fine-tuning απαιτεί αρκετό κόπο και χρόνο όμως αν γίνει σωστά θα μας δώσει ψηλά ποσοστά ακριβείας.

Πριν να πραγματοποιήσουμε την εγχείρηση ενός δικτύου χρειάζεται να ξέρουμε ποιο είναι το όνομα κάθε επιπέδου και ο δείκτης του. Αυτές οι πληροφορίες είναι πολύ χρήσιμες αφού θα χρειαστεί να παγώσουμε και να ξεπαγώσουμε συγκεκριμένα επίπεδα του προ

εκπαιδευμένου δικτύου που θα εκπαιδεύσουμε. Έτσι θα ξέρουμε ποια επίπεδα θα πρέπει να παγώσουμε και να ξεπαγώσουμε ώστε το δίκτυο να μπορεί να μάθει να επιλύει το πρόβλημα που θέλουμε.

Χρησιμοποιούμε τον κώδικα που βρίσκεται στο αρχείο `inspect_model.py` για να δούμε τα επίπεδα και τους αντίστοιχους δείκτες κάθε δικτύου που θα χρησιμοποιήσουμε ώστε να ξέρουμε ποια επίπεδα να παγώσουμε και ποια να ξεπαγώσουμε κατά την εκπαίδευσή τους.

Εκτελούμε το αρχείο με την εντολή: `python inspect_model.py` για να εξετάσουμε την αρχιτεκτονική ολόκληρου του δικτύου ενώ για να παρατηρήσουμε την αρχιτεκτονική του δικτύου χωρίς τα τελευταία πλήρως συνδεδεμένα επίπεδα της χρησιμοποιούμε την εντολή: `python inspect_model.py --include-top -1`. Για να αλλάξουμε την αρχιτεκτονική του δικτύου που εξετάζουμε φορτώνουμε το κατάλληλο δίκτυο από την βιβλιοθήκη Keras. Επίσης, φορτώνουμε τα προεκπαιδευμένα βάρη ImageNet στο δίκτυο που θέλουμε και γράφουμε `include_top = false`, αν θέλουμε να αφαιρέσουμε τα τελευταία πλήρως συνδεδεμένα επίπεδά. Για κάθε επίπεδο του δικτύου τυπώνεται στην οθόνη ο δείκτης του και το όνομα του επιπέδου.

Γνωρίζοντας αυτές τις πληροφορίες θα ξέρουμε που ξεκινούν τα τελευταία πλήρως συνδεδεμένα επίπεδα του δικτύου για να τα αντικαταστήσουμε με τα νέα πλήρως συνδεδεμένα επίπεδα.

Πριν να αντικαταστήσουμε τα τελευταία πλήρως συνδεδεμένα επίπεδα ενός προεκπαιδευμένου δικτύου χρειάζεται να κατασκευάσουμε τα νέα πλήρως συνδεδεμένα επίπεδα που θα βάλουμε στην θέση τους. Δημιουργήσαμε ένα νέο αρχείο με το όνομα `fheadnet.py` μέσα στο μονοπάτι `/home/pyimagesearch/nn/conv` και γράφουμε τον κώδικα που βρίσκεται στο Παράρτημα Γ:

Στις πρώτες 4 γραμμές του κώδικα φορτώνουμε τις βιβλιοθήκες που χρειαζόμαστε.

Ακολουθως, ορίζουμε την κλάση FCHeadNet. Στη συνέχεια, ορίζουμε μια συνάρτηση με το όνομα build που είναι υπεύθυνη για την κατασκευή της αρχιτεκτονικής του δικτύου. Η συνάρτηση αυτή δέχεται τρεις παραμέτρους, την baseModel (το δίκτυο), την classes που είναι το πλήθος των κλάσεων του συνόλου δεδομένων και την D που είναι το πλήθος των κόμβων του πλήρως συνδεδεμένου επιπέδου. Στη συνέχεια, αρχικοποιούμε την παράμετρο headModel που ενώνει το νέο τμήμα του δικτύου με το υπόλοιπο δίκτυο (baseModel.output). Η αρχιτεκτονική του δικτύου μας απαρτίζεται από τα εξής επίπεδα: INPUT => FC => RELU => DO => FC => SOFTMAX. Το τελευταίο πλήρως συνδεδεμένο τμήμα του πρωτότυπου δικτύου VGG16 αποτελείτο από δύο σύνολα 4096 πλήρως συνδεδεμένων επιπέδων. Αντίθετα, το νέο τελευταίο πλήρως συνδεδεμένο τμήμα του δικτύου είναι πολύ πιο απλό. Καταφέραμε να δημιουργήσουμε ένα απλό τελευταίο πλήρες συνδεδεμένο τμήμα που περιέχει λιγότερες παραμέτρους από το προηγούμενο. Όταν υπάρχουν λιγότερες παράμετροι στο τμήμα που προαναφέραμε βοηθούν στην εύκολη εκπαίδευση του δικτύου σε νέες κλάσεις εικόνων. Τέλος, επιστρέφουμε το νέο πλήρως συνδεδεμένο τμήμα που έχουμε κατασκευάσει. Θα εξηγήσουμε σε επόμενη υποενότητα την αντικατάσταση του τελευταίου τμήματος του δικτύου με το τμήμα που μόλις κατασκευάσαμε.

5.6 Εξαγωγή Χαρακτηριστικών με την Χρήση Προ εκπαιδευμένων Δικτύων – Παράδειγμα VGG16

Όπως γνωρίζουμε μπορούμε να χρησιμοποιήσουμε τα συνελκτικά δίκτυα ως κατηγοριοποιητές. Σε αυτή την υποενότητα θα δούμε πως μπορούμε τα χρησιμοποιήσουμε ως εξαγωγείς χαρακτηριστικών.

Η εξαγωγή χαρακτηριστικών πραγματοποιείται με την διακοπή του εμπρόσθιου περάσματος του δίκτυο σε οποιοδήποτε επίπεδο π.χ. επίπεδο συγκέντρωσης, επίπεδο ενεργοποίησης. Ακολουθως, λαμβάνουμε τις τιμές εξόδου του δικτύου στο επίπεδο που

σταματήσαμε και τις χρησιμοποιούμε ως διανύσματα χαρακτηριστικών. Η αυθεντική αρχιτεκτονική του δικτύου VGG16 στην έξοδο, παρέχει τις πιθανότητες μιας εικόνας να ανήκει σε κάθε μια από τις 1000 ετικετοποιημένες κλάσεις του ImageNet. Αφαιρώντας τα πλήρως συνδεδεμένα επίπεδα του VGG16, λαμβάνουμε στην έξοδο το αποτέλεσμα του τελευταίου επιπέδου συγκέντρωσης το οποίο μπορεί να χρησιμοποιηθεί ως διάνυσμα εξαχθέντων χαρακτηριστικών. Επομένως, το τελευταίο επίπεδο του δικτύου είναι ένα επίπεδο συγκέντρωσης με σχήμα $7 \times 7 \times 512$ που υποδηλώνει πως υπάρχουν 512 φίλτρα διαστάσεων 7×7 . Αν συνεχίζαμε το εμπρόσθιο πέρασμα θα λαμβάναμε $512 \times 7 \times 7$ συναρτήσεις ενεργοποίησης. Έτσι, μπορούμε να πάρουμε αυτές τις $7 \times 7 \times 512 = 25088$ τιμές και να τις χειριστούμε ως διάνυσμα εξαχθέντων χαρακτηριστικών. Μπορούμε να επαναλάβουμε την πιο πάνω διαδικασία για ολόκληρα σύνολα δεδομένων στα οποία δεν έχει εκπαιδευτεί το δίκτυο μας (VGG16 στο παράδειγμα αυτό), σχηματίζοντας ένας πίνακα N εικόνων (πλήθος του συνόλου δεδομένων) επί 25088 χαρακτηριστικά (πλήθος εξαχθέντων χαρακτηριστικών) που ποσοτικοποιούν τα περιεχόμενα τους. Λαμβάνοντας τα διανύσματα χαρακτηριστικών. Μπορούμε να εκπαιδεύσουμε μοντέλα μηχανικής μάθησης που λειτουργούν ως κατηγοριοποιητές (όπως Logistic Regression, Random Forest) στηριζόμενοι δίνοντας τους ως είσοδο το διάνυσμα χαρακτηριστικών για να κατηγοριοποιήσουν νέα σύνολα δεδομένων σε νέες κλάσεις. Αξίζει να αναφερθεί πως αποθηκεύουμε τα χαρακτηριστικά που εξήγαμε σε αρχεία στο σύστημα διαχείρισης αρχείων hdf5. Θυμίζουμε πως οι αρχιτεκτονικές συνελκτικών δικτύων μπορούν να κατηγοριοποιήσουν με μεγάλη ακρίβεια, διάφορα σύνολα δεδομένων στα οποία δεν έχουν εκπαιδευτεί αρχικά, με την χρήση του transfer learning.

5.6.1 Εξαγωγή Χαρακτηριστικών Οχημάτων –VGG16

Σε αυτήν την υποενότητα θα μελετήσουμε την διαδικασία εξαγωγής χαρακτηριστικών όπως αυτή πραγματοποιήθηκε στις εφαρμογές της αναγνώρισης και ανίχνευσης οχημάτων μας. Στην συγκεκριμένη υποενότητα θα περιγράψουμε την διαδικασία που ακολουθήσαμε για εξαγωγή χαρακτηριστικών οχημάτων με το δίκτυο VGG16.

Αρχικά, φορτώνουμε τις βιβλιοθήκες της Python που θα χρειαστούμε. Στη συνέχεια, φορτώνουμε την αρχιτεκτονική του προ εκπαιδευμένου δικτύου που θα χρησιμοποιήσουμε για την εξαγωγή των χαρακτηριστικών. Στις εφαρμογές μας, εξαγάγαμε χαρακτηριστικά με τα διαφορετικά συνελκτικά δίκτυα και χρησιμοποιήσαμε διαφορετικές διαστάσεις εικόνων εισόδου σε πολλά από αυτά. Για να μετατρέψουμε τις εικόνες εισόδου σε εικόνες με ίδιες διαστάσεις (π.χ. 224x224 pixels, 192x192 pixels, 197x197 pixels, 32x32pixels), χρησιμοποιήσαμε τις συναρτήσεις ImagePreprocessor & Aspect-aware Preprocessor, όπως περιγράψαμε και στο προηγούμενο κεφάλαιο. Ακολούθως, χρησιμοποιούμε την κλάση LabelEncoder για να μετατρέψει τις κλάσεις μας από λέξεις σε αριθμούς. Φορτώνουμε την συνάρτηση που θα μας βοηθήσει να αποθηκεύσουμε το διάνυσμα χαρακτηριστικών που θα εξάξουμε. Στην γραμμή εντολών δίνουμε δύο παραμέτρους, το σύνολο των δεδομένων - διεύθυνση του συνόλου δεδομένων και την έξοδο – διεύθυνση στην οποία θα αποθηκευτεί το διάνυσμα χαρακτηριστικών που θα παράξουμε. Εφαρμόζουμε ένα batch size μεγέθους 32 . Ακολούθως, ανακατεύουμε τις διευθύνσεις των εικόνων για ευκολότερο διαχωρισμό του συνόλου δεδομένων εκπαίδευσης και του συνόλου επαλήθευσης και κωδικοποιούμε τις κλάσεις. Έπειτα, λαμβάνουμε την κλάση μιας εικόνας από την διεύθυνση. Φορτώνουμε το προ εκπαιδευμένο δίκτυο μας (φορτώνουμε τα βάρη ImageNet και ρυθμίζουμε τις διαστάσεις εισόδου του και αφαιρούμε το τελευταία πλήρως συνδεδεμένα επίπεδά του. Όπως περιγράψαμε στην προηγούμενη υποενότητα, θα σταματήσουμε το εμπρόσθιο πέρασμα στο επίπεδο που βρίσκεται πριν το τελευταίο πλήρως συνδεδεμένο επίπεδο του δικτύου. Αρχικοποιούμε τον hdf5 εγγραφέα που θα αποθηκεύσει τα χαρακτηριστικά σε ένα αρχείο. Η πρώτη παράμετρος του είναι οι διαστάσεις του συνόλου δεδομένων μας που είναι το πλήθος των εικόνων που βρίσκονται σε αυτό και η δεύτερη παράμετρος είναι το μέγεθος του διανύσματος χαρακτηριστικών που είναι $512 \times 7 \times 7 = 25088$. Ακολούθως, αποθηκεύουμε τα κωδικοποιημένα ονόματα των κλάσεων. Αρχικοποιούμε την μπάρα προόδου που μας δείχνει πότε θα ολοκληρωθεί η εξαγωγή χαρακτηριστικών. Δίνουμε τις εικόνες στο δίκτυο ανά batches, εξάγοντας το μονοπάτι της διεύθυνσης και τις αντίστοιχες

ετικέτες για το συγκεκριμένο batch. Αποθηκεύουμε τις εικόνες σε μια λίστα πριν να τις δώσουμε στο δίκτυο.). Περνούμε πάνω από κάθε εικόνα στο batch, την φορτώνουμε από τον δίσκο και την μετατρέπουμε σε ένα πίνακα Keras. Μεταβάλλουμε το μέγεθος της κάθε εικόνας ώστε όλες οι εικόνες να έχουν το ίδιο μέγεθος (224x224) και πραγματοποιούμε την κατάλληλη προ επεξεργασία σε αυτές (αυξάνουμε τις διαστάσεις τους). Προσθέτουμε κάθε εικόνα στο batch. Δίνουμε τις εικόνες στο δίκτυο, χρησιμοποιώντας την συνάρτηση `vstack` από την βιβλιοθήκη `NumPy`. Η συνάρτηση αυτή κολλάει τις εικόνες μεταξύ τους κάθετα ώστε το σχήμα τους να είναι $(N, 224, 224, 3)$, όπου το N είναι το μέγεθος του batch, οι επόμενες δύο διαστάσεις είναι το μέγεθος της εικόνας σε πλάτος επί μήκος και η τελευταία διάσταση αναπαριστά τα κανάλια χρωμάτων στον χώρο αναπαράστασης χρωμάτων RGB. Οι τιμές του τελευταίου επιπέδου εξάγονται από το τελευταίο επίπεδο συγκέντρωση του δικτύου VGG16.

Η έξοδος του τελευταίου επιπέδου συγκέντρωσης, που είναι επίπεδο συγκέντρωσης `max pooling` δύο διαστάσεων, είναι σχήματος $(N, 512, 7, 7)$, υποδηλώνοντας πως υπάρχουν 512 φίλτρα μεγέθους 7×7 . Για να χειριστούμε αυτές τις τιμές ως διάνυσμα χαρακτηριστικών χρειάζεται να τις `flatten` σε ένα πίνακα δύο διαστάσεων μεγέθους $(N, 25088)$, χρησιμοποιώντας την συνάρτηση `reshape` στα χαρακτηριστικά. Στην συνάρτηση αυτή περνούμε ως πρώτη παράμετρο το N μέγεθος του batch και ως δεύτερη παράμετρο το μέγεθος του διανύσματος των χαρακτηριστικών δηλαδή το μέγεθος του τελευταίου επιπέδου συγκέντρωσης. Γράφουμε τα χαρακτηριστικά και τις ετικέτες σε ένα σύνολο δεδομένων τύπου HDF5. Τέλος, κλείνουμε, τους καταγραφείς δεδομένων του `hdf5`.

Πραγματοποιήσαμε την πιο πάνω διαδικασία εξαγωγής χαρακτηριστικών για κάθε δίκτυο που χρησιμοποιήσαμε με την διαφορά ότι σε κάθε δίκτυο αλλάξαμε το μέγεθος της συνάρτησης `flatten` και τις διαστάσεις των δεδομένων που αποθηκεύονται από τον καταγραφέα ώστε να συμφωνεί με τις διαστάσεις του διανύσματος χαρακτηριστικών που λαμβάνουμε στην έξοδο. Αυτό συμβαίνει λόγω του διαφορετικού πλήθους επιπέδων και

των διαφορετικών τύπων τους στις αρχιτεκτονικές συνελκτικών δικτύων που χρησιμοποιήσαμε.

5.6.1 Εξαγωγή Χαρακτηριστικών Οχημάτων – ResNet50

Όσον αφορά την εξαγωγή χαρακτηριστικών οχημάτων με το συνελκτικό δίκτυο ResNet50 ακολουθούμε την διαδικασία που έχουμε περιγράψει και έχουμε διεξάγει για το VGG16 αλλά με μερικές διαφορές. Θυμίζουμε πως το σχήμα της εισόδου του δικτύου ήταν (224, 224, 3). Χρησιμοποιώντας, το δίκτυο ResNet50 για εξαγωγή των χαρακτηριστικών των οχημάτων, για εικόνες μεγέθους 224x224 pixels, σταματούμε την εκπαίδευση στο τελευταίο επίπεδο συγκέντρωσης που χρησιμοποιεί την συνάρτηση average pooling. Το επίπεδο αυτό είναι διαστάσεων (N, 100352). Έτσι, αρχικοποιούμε τον hdf5 εγγραφέα που θα αποθηκεύσει τα χαρακτηριστικά σε ένα αρχείο, θέτοντας την η δεύτερη παράμετρό του, που είναι το μέγεθος του διάνυσματος χαρακτηριστικών, να είναι 100352. Η έξοδος του τελευταίου επιπέδου συγκέντρωσης, που είναι επίπεδο συγκέντρωσης average pooling δύο διαστάσεων, είναι σχήματος (N, 2048, 7, 7), υποδηλώνοντας πως υπάρχουν 2048 φίλτρα μεγέθους 7x7. Για να χειριστούμε αυτές τις τιμές ως διάνυσμα χαρακτηριστικών χρειάζεται να τις flatten σε ένα πίνακα δύο διαστάσεων μεγέθους (N, 100352), χρησιμοποιώντας την συνάρτηση reshape στα χαρακτηριστικά.

5.6.3 Εξαγωγή Χαρακτηριστικών Οχημάτων – MobileNet

Όσον αφορά την εξαγωγή χαρακτηριστικών οχημάτων με το συνελκτικό δίκτυο MobileNet ακολουθούμε την διαδικασία που έχουμε περιγράψει και έχουμε διεξάγει για τα υπόλοιπα δίκτυα αλλά με μερικές διαφορές. Συγκεκριμένα, χρησιμοποιώντας το δίκτυο MobileNet για εξαγωγή των χαρακτηριστικών των οχημάτων, για εικόνες μεγέθους 224x224 pixels (είσοδος , σταματούμε την εκπαίδευση στο τελευταίο επίπεδο συγκέντρωσης που χρησιμοποιεί την συνάρτηση average pooling. Το επίπεδο αυτό είναι διαστάσεων (N, 1024,

7, 7). Έτσι, αρχικοποιούμε τον hdf5 εγγραφέα που θα αποθηκεύσει τα χαρακτηριστικά σε ένα αρχείο, θέτοντας την η δεύτερη παράμετρό του, που είναι το μέγεθος του διανύσματος χαρακτηριστικών, να είναι 50176 (1024x7x7). Η έξοδος του τελευταίου επιπέδου συγκέντρωσης, που είναι επίπεδο συγκέντρωσης average pooling δύο διαστάσεων, είναι σχήματος (N, 1024, 7, 7), υποδηλώνοντας πως υπάρχουν 1024 φίλτρα μεγέθους 7x7. Για να χειριστούμε αυτές τις τιμές ως διάνυσμα χαρακτηριστικών χρειάζεται να τις flatten σε ένα πίνακα δύο διαστάσεων μεγέθους (N, 50176), χρησιμοποιώντας την συνάρτηση reshape στα χαρακτηριστικά.

5.7 Δημιουργία Δειγμάτων Εκπαίδευσης και Ελέγχου για DCNNs

5.7.1 Δημιουργία Δειγμάτων Εκπαίδευσης και Ελέγχου για Προεκπαιδευμένων Εκδόσεων Συνελικτικών Δικτύων - ResNet50, VGG16, ShallowNet, MobileNet

Δημιουργούμε ένα καινούργιο αρχείο με το όνομα train_model_VGG16_pc.py. Ακολούθως, φορτώνουμε τις κατάλληλες βιβλιοθήκες. Στην συνέχεια, κατασκευάζουμε τον αναλυτή των παραμέτρων που λαμβάνουμε από την γραμμή εντολών. Συγκεκριμένα λαμβάνουμε δύο παραμέτρους, την διεύθυνση του αρχείου των χαρακτηριστικών των οχημάτων που έχουμε εξάγει και των ετικετών των δύο κατηγοριών. βρίσκεται μετά τον διακόπτη - - db και την διεύθυνση αποθήκευσης των δεδομένων εξόδου που εισάγουμε μετά τον διακόπτη - - model. Έπειτα, ανοίγουμε την βάση των χαρακτηριστικών οχημάτων που βρίσκεται σε μορφή HDF5 αρχείου για ανάγνωση. Αμέσως μετά, θέτουμε τον δείκτη σύμφωνα με τον οποίο θα χωρίσουμε τα χαρακτηριστικά της βάσης δεδομένων σε σύνολο δεδομένων εκπαίδευσης και σύνολο δεδομένων επαλήθευσης, νοουμένου ότι τα δεδομένα ήταν ήδη αποθηκευμένα με τυχαία σειρά στον δίσκο. Η αναλογία σύμφωνα με την οποία αποφασίσαμε να χωρίσουμε τα δεδομένα μας είναι 75%-25%. Βάσει αυτής, το 75% του συνόλου δεδομένων θα χρησιμοποιηθούν στην εκπαίδευση του δικτύου ενώ το υπόλοιπο 25% στην επαλήθευσή του. Γνωρίζουμε εκ των προτέρων το πλήθος των

εγγραφών που υπάρχουν μέσα στο σύνολο δεδομένων μας και γι' αυτό τον λόγο υπολογίζουμε το 75% του δείκτη i της βάσης δεδομένων. Τα δεδομένα που βρίσκονται πριν τον δείκτη αποτελούν το σύνολο δεδομένων/χαρακτηριστικών εκπαίδευσης ενώ όσα βρίσκονται μετά από αυτόν, ανήκουν στο σύνολο δεδομένων/χαρακτηριστικών επαλήθευσης.

Ακολουθούμε την πιο πάνω διαδικασία για τις προεκπαιδευμένες εκδόσεις των συνελκτικών δικτύων, VGG16 Transfer Learning και MobileNet Transfer Learning με μόνη διαφορά την αλλαγή του μεγέθους των εικόνων εισόδου από 224x224pixels σε 192x192 pixels και στους δύο τύπους δικτύων μετά από αξιολόγηση των αποτελεσμάτων εκπαίδευσης και επαλήθευσης των κατηγοριοποιητών οχημάτων.

Στη συνέχεια, μεταβάλλουμε το μέγεθος των εικόνων εισόδου σε 197x197 pixels όσον αφορά το δίκτυο ResNet50 και 192x192pixels τα δίκτυα VGG16 και MobileNet, αξιολογώντας πρώτα τα αποτελέσματα των κατηγοριοποιητών με μέγεθος εικόνων εισόδου 224x224 pixels πρώτα εκπαιδεύουμε τους κατηγοριοποιητές με τις προαναφερθείσες ρυθμίσεις. Όσον αφορά το δίκτυο MobileNet, το έχουμε εκπαιδεύσει και με εικόνες μεγέθους 160x160pixels και 128x128pixels πραγματοποιώντας τις κατάλληλες αλλαγές στις διαστάσεις των εικόνων, ώστε να διατηρηθεί το aspect ratio τους, όπως ακριβώς περιγράψαμε πιο πάνω.

5.7.6 Δημιουργία Δειγμάτων Εκπαίδευσης και Ελέγχου για Προεκπαιδευμένες Transfer Learning Εκδόσεις Συνελκτικών Δικτύων - VGG16 Transfer Learning και MobileNet Transfer Learning

Δημιουργούμε ένα καινούργιο αρχείο με το όνομα `train_model_VGG16_pc.py`. Ακολούθως, φορτώνουμε τις κατάλληλες βιβλιοθήκες. Στην συνέχεια, κατασκευάζουμε τον αναλυτή των παραμέτρων που λαμβάνουμε από την γραμμή εντολών. Συγκεκριμένα

λαμβάνουμε δύο παραμέτρους την διεύθυνση του αρχείου του συνόλου δεδομένων εισόδου που βρίσκεται μετά τον διακόπτη `-db` και την διεύθυνση αποθήκευσης των δεδομένων εξόδου, μετά την εκπαίδευση με τον αλγόριθμο Logistic Regression. Έπειτα, ανοίγουμε το σύνολο δεδομένων εισόδου που βρίσκεται σε μορφή HDF5 αρχείου για ανάγνωση. Σύμφωνα με αυτό, αποφασίζουμε να χωρίσουμε τα δεδομένα μας με τον κανόνα 75:25. Βάσει αυτού του κανόνα, το 75% του συνόλου δεδομένων θα χρησιμοποιηθούν στην εκπαίδευση του δικτύου ενώ το υπόλοιπο 25% στην επαλήθευσή του.

Έχουμε περιγράψει σε προηγούμενο κεφάλαιο τα χαρακτηριστικά του συνόλου δεδομένων που χρησιμοποιήσαμε. Στην παρούσα υπενότητα, θα περιγράψουμε την διαδικασία που κάναμε για να δημιουργήσουμε τα δείγματα εκπαίδευσης και ελέγχου όσον αφορά την εκπαίδευση ενός προεκπαιδευμένου δικτύου στην κατηγοριοποίηση οχημάτων. Σε αυτές τις εφαρμογές, πραγματοποιούμε, Data Augmentation κατά την εκπαίδευση του δικτύου. Οι λόγοι που χρησιμοποιούμε data augmentation είναι για να αποφύγουμε την υπερεκπαίδευση, να αυξήσουμε το ποσοστό ακριβείας και να βοηθήσουμε το δίκτυο να μάθει τις κλάσεις παρόλο που το σύνολο δεδομένων μας είναι μικρό (700-800 εικόνες ανά κλάση και αποτελείται από 2 κλάσεις). Το σύνολο δεδομένων μας βρίσκεται στο μονοπάτι: `/Data/drone_dataset/train_p_n/equal_size/{class_name}/example.jpg`. Επομένως, παίρνουμε τα μονοπάτια των εικόνων που βρίσκονται στον δίσκο και εξάγουμε τις ετικέτες των κλάσεων από αυτά.

Αρχικοποιούμε τους προεπεξεργαστές εικόνων. Θα μεταβάλουμε το μέγεθος των εικόνων εισόδου έτσι ώστε όλες να έχουν 224x224 pixels (το απαιτούμενο μέγεθος εισόδου των δικτύων VGG16 Transfer Learning, MobileNet Transfer Learning) ενώ ταυτόχρονα διατηρούμε το αρχικό aspect ratio της εικόνας. Στην συνέχεια, εφαρμόζουμε τους προεπεξεργαστές εικόνας για να φορτώσουμε τα δεδομένα και τις ετικέτες από τον δίσκο. Ακολούθως, δημιουργούμε τα σύνολα δεδομένων εκπαίδευσης και ελέγχου

χρησιμοποιώντας τον κανόνα 75:25. Σύμφωνα με αυτόν τον κανόνα, το 75% του συνόλου δεδομένων χρησιμοποιείται για εκπαίδευση ενώ το υπόλοιπο 25% για επαλήθευση. Αξίζει να αναφερθεί πως στην παράμετρο `random_state` δίνουμε την τιμή 42 ώστε τα δεδομένα μας να χωριστούν με τυχαίο τρόπο. Τέλος κωδικοποιούμε τις ετικέτες με την χρήση του `one hot encoding` από αριθμούς σε αριθμητικά διανύσματα.

Ακολουθούμε την πιο πάνω διαδικασία για τις προεκπαιδευμένες εκδόσεις των συνελικτικών δικτύων, VGG16 Transfer Learning και MobileNet Transfer Learning με μόνη διαφορά την αλλαγή του μεγέθους των εικόνων εισόδου από 224x224pixels σε 192x192 pixels,...κλπ και στους δύο τύπους δικτύων μετά από αξιολόγηση των αποτελεσμάτων εκπαίδευσης και επαλήθευσης των κατηγοριοποιητών οχημάτων.

5.8 Εγχείρηση Συνελικτικών Δικτύων

Εγχείρηση Συνελικτικών Δικτύων Προεκπαιδευμένων Εκδόσεων Συνελικτικών Δικτύων VGG16 Transfer Learning και MobileNet Transfer Learning.

Στην υποενότητα αυτή, θα εξηγήσουμε πως πραγματοποιούμε την εγχείρηση συνελικτικών δικτύων στις προεκπαιδευμένες Transfer Learning εκδόσεις τους. Στο αρχείο `finetune_flowers17.py` που έχουμε ήδη δημιουργήσει φορτώνουμε ένα συνελικτικό δίκτυο π.χ VGG16 από τον δίσκο και τα προεκπαιδευμένα βάρη ImageNet. Θέτουμε το μέγεθος των δεδομένων εισόδου να είναι σχήματος 224, 224, 3 ενώ παράλληλα δεν φορτώνουμε τα τελευταία πλήρως συνδεδεμένα επίπεδά του. Στην θέση τους τοποθετούμε το τμήμα τμήμα FCHeadNet που θα είναι το νέο τελευταίο πλήρως συνδεδεμένο τμήμα του δικτύου. Το νέο τελευταίο τμήμα του δικτύου αποτελείται από πολλά πλήρως συνδεδεμένα επίπεδα και ένα επίπεδο softmax κατηγοριοποίησης. Στην συνέχεια, δίνουμε στο νέο τελευταίο τμήμα του δικτύου, το υπόλοιπο δίκτυο, το πλήθος των κλάσεων/ετικετών κατηγοριοποίησης ($n=2$) και το πλήθος των κόμβων ($m=256$) ενός πλήρως συνδεδεμένου

επιπέδου. Η πραγματική εγχείρηση δικτύου συμβαίνει όταν τοποθετούμε το υπόλοιπο δίκτυο ως είσοδο και το νέο τμήμα του δικτύου ως την έξοδο του μοντέλου. Όμως, το δίκτυο δεν είναι έτοιμο για εκπαίδευση αφού χρειάζεται να παγώσουμε μερικά από τα βάρη του ώστε να μην ενημερωθούν κατά την εκτέλεση του αλγορίθμου ανάστροφης μετάδοσης σφάλματος. Θα δούμε πως γίνεται αυτή η διαδικασία σε άλλη ενότητα.

Επαναλαμβάνουμε την πιο πάνω διαδικασία για το ίδιο δίκτυο όμως θέτουμε το σχήμα δεδομένων εισόδου να είναι διαστάσεων 197, 197, 3. Ακολούθως, επαναλαμβάνουμε ξανά την προαναφερθείσα διαδικασία φορτώνοντας το δίκτυο MobileNet και θέτουμε το σχήμα των δεδομένων εισόδου να είναι διαστάσεων 224,224, 3 και 192,192,3. Η χρήση του δικτύου MobileNet στην διαδικασία αυτή μας οδηγεί στην αλλαγή του πλήθους των κόμβων που θα βάλουμε σε ένα πλήρως συνδεδεμένο επίπεδο του δικτύου. Συγκεκριμένα αντί 256 κόμβους θα βάλουμε 512. Συνήθως, το πλήθος των κόμβων στο πλήρως συνδεδεμένο επίπεδο είναι μικρότερο ή ίσο με το πλήθος των φίλτρων που υπάρχουν στο προηγούμενο επίπεδο συγκέντρωσης.

5.9 Φάση Εκπαίδευσης, Ελέγχου και Επαλήθευσης DCNNs

5.9.1 Φάση Εκπαίδευσης, Ελέγχου και Επαλήθευσης Προεκπαιδευμένης Έκδοσης Συνελικτικού Δικτύου.

5.9.1.1 Εκπαίδευση Logistic και Επαλήθευση Regression Κατηγοριοποιητή με GridSearch

Θα εκπαιδεύσουμε έναν κατηγοριοποιητή που χρησιμοποιεί GridSearchCV μαζί με Logistic Regression. Ορίζουμε τις παραμέτρους που θέλουμε να ρυθμίσουμε και ξεκινούμε μια αναζήτηση πλαισίου. Στην αναζήτηση πλαισίου θα αξιολογήσουμε το μοντέλο μας για κάθε τιμή της παραμέτρου C. Οι τιμές της C που χρησιμοποιούμε είναι 0.0001, 0.001, 0.01, 0.1, 1.0. Μόλις βρούμε την καλύτερη τιμή της παραμέτρου C, παράγουμε μια αναφορά κατηγοριοποίησης για το σύνολο ελέγχου. Τέλος, αποθηκεύουμε το μοντέλο στον δίσκο.

Επαληθεύουμε τον έναν τέτοιο κατηγοριοποιητή, δίνοντας του να κατηγοριοποιήσει το σύνολο εικόνων επαλήθευσης μετά την εκπαίδευση. Το αποτέλεσμα που βλέπουμε στην οθόνη, περιλαμβάνει πίνακα ποσοστών για τις μετρικές μας και σε μερικά από τα δίκτυα ένα συνολικό σκορ του κατηγοριοποιητή.

Επαναλαμβάνουμε την προαναφερθείσα διαδικασία, με μόνες διαφορές, την φόρτωση των δικτύων ResNet50, ShallowNet και MobileNet και την παροχή των αντίστοιχων χαρακτηριστικών οχημάτων ως είσοδό τους (που εξάχθηκαν κατά την διαδικασία εξαγωγής χαρακτηριστικών).

5.9.2 Φάση Εκπαίδευσης, Ελέγχου και Επαλήθευσης Προεκπαιδευμένων Εκδόσεων Transfer Learning Συνελικτικών Δικτύων - Transfer Learning VGG16 και Transfer Learning MobileNet

Περνούμε πάνω από όλα τα επίπεδα του πρωτότυπου δικτύου και τα παγώνουμε ώστε τα βάρη τους να μην ενημερώνονται κατά την διαδικασία εκπαίδευσης. Ακολουθώντας, αρχικοποιούμε τον αλγόριθμο ενημέρωσης βαρών RMSprop με ρυθμό μάθησης 0.001. Αρχικά, χρησιμοποιούμε πολύ μικρή τιμή στον ρυθμό μάθησης (όπως αυτήν που χρησιμοποιήσαμε). Αφού ρυθμίσαμε τα επίπεδα του δικτύου μας με τέτοιο τρόπο ώστε να μην εκπαιδεύονται (να μην μεταβάλλονται οι τιμές των βαρών τους), μεταγλωττίζουμε το μοντέλο με κύρια μετρική την ακρίβεια και τον τύπο του σφάλματος να είναι `categorical_crossentropy`. Στη συνέχεια, εκπαιδεύουμε το νέο τελευταίο πλήρως συνδεδεμένο τμήμα του δικτύου με την χρήση της `data augmentation` συνάρτησης για 25 εποχές (συνήθως το νέο τμήμα του δικτύου εκπαιδεύεται για 10-30 εποχές αναλόγως του συνόλου δεδομένων), ενώ κρατούμε παγωμένα τα υπόλοιπα επίπεδα του δικτύου. Αυτό θα επιτρέψει στα νέα πλήρως συνδεδεμένα επίπεδα του τελευταίου τμήματος του δικτύου

να μάθουν δηλαδή τα βάρη τους να ενημερωθούν με νέες τιμές κατά την διαδικασία της μάθησης. Θυμίζουμε πως ενώ κάθε εικόνα μεταφέρεται προς τα μπροστά από το επίπεδο εισόδου στο επίπεδο εξόδου του δικτύου, οι κλίσεις μεταφέρονται προς τα πίσω μέχρι τα πλήρως συνδεδεμένα επίπεδα καθώς ο στόχος μας είναι να ζεστάνουμε το νέο τελευταίο πλήρως συνδεδεμένο τμήμα του δικτύου.

Έπειτα, αξιολογούμε την απόδοση του δικτύου στο σύνολο δεδομένων επαλήθευσης και παράγουμε μια αναφορά κατηγοριοποίησης. Ο κώδικας μας είναι γραμμένος με τέτοιο τρόπο ώστε να έχουμε την δυνατότητα να συγκρίνουμε τις συνέπειες του fine-tuning πριν και μετά την εκπαίδευση του τελευταίου νέου πλήρως συνδεδεμένου τμήματος του δικτύου. Έχουμε ήδη αρχικοποιήσει και εκπαιδεύσει μερικώς το τελευταίο πλήρως συνδεδεμένο τμήμα του δικτύου, και τώρα ξεπαγώνουμε το τελευταίο σύνολο επιπέδων συνέλιξης και τα ρυθμίζουμε ώστε να μπορούν να εκπαιδευτούν. Ακολουθώντας, επαναμεταγλωττίζουμε το μοντέλο για να λάβουν χώρα οι αλλαγές που πραγματοποιήσαμε. Αυτή την φορά, αφού έχουμε ζεστάνει το δίκτυο, χρησιμοποιούμε τον αλγόριθμο ενημέρωσης συναπτικών βαρών στοχαστικής κατάβασης κλίσης με μικρό ρυθμό μάθησης (0.001). Στη συνέχεια, εκπαιδεύουμε το δίκτυο μας για περισσότερες από 100 εποχές αφήνοντας περισσότερο χρόνο προσαρμογής στα φίλτρα συνέλιξης. Έτσι, τα φίλτρα συνέλιξης μαθαίνουν τα μοτίβα που βρίσκονται στο σύνολο δεδομένων μας. Σε αυτό το σημείο μπορούμε να αξιολογήσουμε το δίκτυο καθώς και να αποθηκεύσουμε τα βάρη του στον δίσκο.

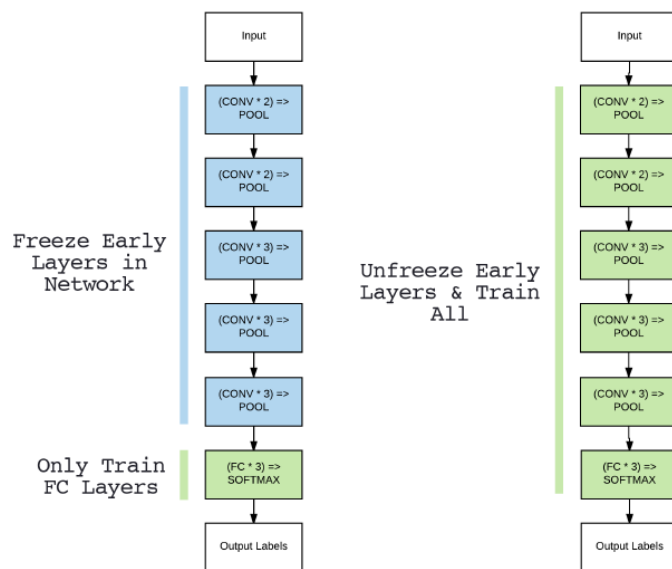
Επαληθεύουμε τον έναν κατηγοριοποιητή, δίνοντας του να κατηγοριοποιήσει το σύνολο εικόνων επαλήθευσης μετά την εκπαίδευση. Το αποτέλεσμα που βλέπουμε στην οθόνη, περιλαμβάνει πίνακα ποσοστών και σε μερικά από τα δίκτυα ένα συνολικό σκορ του κατηγοριοποιητή.

Οι αρχιτεκτονικές βαθιών συνελκτικών δικτύων που έχουν περισσότερα επίπεδα (π.χ VGG16) από αυτές των απλών συνελκτικών, διαθέτουν επίσης περισσότερες παραμέτρους

σε σχέση με τα υπόλοιπα δίκτυα. Στις περιπτώσεις αυτές, ξεπαγώνουμε μόνο τα τελευταία επίπεδα συνέλιξης και συνεχίζουμε την εκπαίδευση. Αν βελτιωθεί η ακρίβεια κατηγοριοποίησης και συνεχίζει να βελτιώνεται χωρίς να συμβαίνει το φαινόμενο της υπερεκπαίδευσης, μπορούμε να ξεπαγώσουμε περισσότερα επίπεδα προς εκπαίδευση.

Βλέπουμε πως μπορούμε να εκπαιδεύσουμε αρχιτεκτονικές δικτύων όπως τα δίκτυα VGG16, ResNet50, ShallowNet και MobileNet σε χαρακτηριστικά οχημάτων και να παράξουμε τα μοντέλα κατηγοριοποίησης οχημάτων. Έπειτα, θα τα δώσουμε στους ανιχνευτές οχημάτων για να ανιχνεύσουν τα οχήματα που βρίσκονται μέσα στις εικόνες που τους δίνουμε ως είσοδο.

5.10



Σχήμα 5.5 – Transfer Learning Freeze/Unfreeze Network Layers

Ανίχνευση Οχημάτων με την χρήση Συνελικτικών Δικτύων και Υλοποίηση της

5.10.1 Υλοποίηση Sliding Window

Δημιουργούμε ένα νέο αρχείο με το όνομα `simple_obj_det_rc.py`. Στη συνέχεια, αποθηκεύουμε το αρχείο αυτό μέσα στον υποφάκελο `utils`.

Η συνάρτηση sliding window λαμβάνει τρεις παραμέτρους. Η πρώτη παράμετρος (image), είναι η εικόνα εισόδου. Η δεύτερη παράμετρος είναι το step, που είναι το μέγεθος του βήματος που κάνει το sliding window. Η παράμετρος step δείχνει πόσα pixels θα αγνοήσουμε στην x και στην y κατεύθυνση. Η παράμετρος step καθορίζεται σύμφωνα με το σύνολο δεδομένων και ρυθμίζεται ώστε να δίνει την καλύτερη επίδοση στις εικόνες του συνόλου δεδομένων. Οι πιο δημοφιλείς τιμές της παραμέτρου αυτής είναι 4, 8 και 16 pixels.

Όσο πιο μικρό είναι το μέγεθος του βήματος τόσο πιο πολλά παράθυρα θα εξετάσουμε. Ενώ όσο πιο μεγάλο είναι το μέγεθος του βήματος, τόσο πιο λίγα παράθυρα εξετάζουμε. Αν και η δεύτερη επιλογή είναι πιο φτηνή υπολογιστικά, εντούτοις υπάρχει η πιθανότητα να μην ανιχνεύσει όλα τα αντικείμενα που βρίσκονται μέσα στην εικόνα.

Η τελευταία παράμετρος που λαμβάνει η συνάρτηση είναι η ws. Και καθορίζει το πλάτος και το μήκος του παραθύρου που εξάγουμε από την εικόνα σε pixels. Ακολούθως, μετακινούμε το παράθυρο με την χρήση δύο επαναλήψεων που περνούν πάνω από τις (x,y) συντεταγμένες της εικόνας, αυξάνοντας τους μετρητές των κατευθύνσεων x,y κατά step size. Τέλος, επιστρέφουμε τις συντεταγμένες (x,y) του sliding window μαζί με την περιοχή ενδιαφέροντος. Η συγκεκριμένη περιοχή ενδιαφέροντος, στην συνέχεια θα δοθεί ως είσοδος σε ένα συνελκτικό δίκτυο για κατηγοριοποίηση.

5.10.2 Υλοποίηση Image Pyramid

Μια image pyramid είναι μια αναπαράσταση μιας εικόνας σε πολλές και διαφορετικές κλίμακες. Ο συνδυασμός της image pyramid και του sliding window οδηγεί στον εντοπισμό διαφορετικής κλίμακας οχημάτων που βρίσκονται σε διαφορετικές τοποθεσίες μέσα στις εικόνες. Για παράδειγμα, εφαρμόζουμε το sliding window πάνω στην πρωτότυπη εικόνα εισόδου. Στη συνέχεια, παρατηρούμε πως το αντικείμενο είναι πολύ μεγάλο και δεν

χωράει μέσα στο sliding window, υποδεικνύοντας πως δεν θα ανιχνευτεί όταν αυτό περάσει από πάνω του. Παρόλα αυτά, εφαρμόζοντας την image pyramid, μειώνουμε την κλίμακα της εικόνας και του αντικειμένου και έτσι το sliding window μπορεί να περάσει ακριβώς πάνω από το αντικείμενο, που δεν είναι πλέον μεγαλύτερο σε μέγεθος από αυτό, και πιθανότατα να το εντοπίσει.

Δημιουργούμε μια συνάρτηση με το όνομα image_pyramid μέσα στο αρχείο simple_obj_det.py. και γράφουμε τον κώδικα της συνάρτησης αυτής. Η προαναφερθείσα συνάρτηση, χρειάζεται τρεις παραμέτρους την image, την scale και την minSize. Η παράμετρος image είναι η εικόνα που λαμβάνει η συνάρτηση. Η scale αναφέρεται στην κλίμακα που καθορίζει πόσο θα μεταβληθεί το μέγεθος της εικόνας σε κάθε επίπεδο της πυραμίδας. Μικρή τιμή στην παράμετρο scale, δημιουργεί περισσότερα επίπεδα στην image pyramid ενώ μεγαλύτερη τιμή λιγότερα. Η συνήθης τιμή της παραμέτρου scale είναι μεταξύ 1.1 - 1.5 με αυξομειώσεις του 0.1 ή 0.05. Η παράμετρος minSize, είναι η ελάχιστες διαστάσεις (πλάτος, μήκος) της εικόνας που απαιτούνται σε ένα επίπεδο. Αν το μέγεθος μιας εικόνας γίνει μικρότερο από την τιμή της παραμέτρου minSize, τότε δεν κατασκευάζεται άλλο επίπεδο στην image pyramid. Ακολούθως, επιστρέφουμε την πρωτότυπη εικόνα που βρίσκεται στην βάση της πυραμίδας. Στη συνέχεια, παίρνουμε μέσα από κάθε επίπεδο της πυραμίδας. Υπολογίζουμε το μέγεθος της εικόνας στο επόμενο επίπεδο της πυραμίδας διατηρώντας πάντα το aspect ratio της.

Η κλίμακα ρυθμίζεται από τον παράγοντα κλίμακας. Ακολούθως, ελέγχουμε το μέγεθος της εικόνας που μόλις έχει δημιουργηθεί σε ένα καινούργιο επίπεδο. Αν αυτό είναι μικρότερο από το ελάχιστο δυνατό, τότε σταματούμε την κατασκευή καινούργιων επιπέδων στην πυραμίδα. Τέλος, επιστρέφουμε την εικόνα που βρίσκεται στο παρόν επίπεδο της image pyramid στο πελάτη που κάλεσε την συνάρτηση image_pyramid

Συνδυάζοντας τις συναρτήσεις sliding window και image pyramid μπορούμε να ανιχνεύσουμε οχήματα σε διαφορετικές τοποθεσίες και κλίμακες μέσα σε μια εικόνα.

5.10.3 Υλοποίηση Batch Processing

Μέχρι στιγμής εξηγήσαμε πως εφαρμόζουμε πυραμίδες εικόνων και μετακινούμενα παράθυρα για την εξαγωγή συγκεκριμένων περιοχών ενδιαφέροντος (ROIs) χωρίς να έχουμε εξηγήσει πως μπορούμε να προβλέψουμε την ετικέτα της κλάσης κάθε περιοχής ενδιαφέροντος. Αυτό μπορεί να γίνει με την χρήση της συνάρτησης `predict` του Keras πάνω στο μοντέλο μας όπως ακριβώς κάναμε και στην κατηγοριοποίηση οχημάτων από εικόνες.

Τα συνελκτικά δίκτυα είναι πιο αποδοτικά με την όταν επεξεργάζονται τα δεδομένα ανά batch. Επομένως, θα κατασκευάσουμε μια συνάρτηση, την `classify_batch`, μέσα στο αρχείο `simple_object_det.py` που λαμβάνει ένα σύνολο (batch) περιοχών ενδιαφέροντος και ένα σύνολο (batch) συντεταγμένων (x,y), κάνει προβλέψεις σύμφωνα με αυτά και επιστρέφει τις ετικέτες και τις πιθανότητες των κλάσεων/κατηγοριών με την περισσότερη εμπιστοσύνη.

Η συνάρτηση `classify_batch` λαμβάνει πολλές παραμέτρους ως είσοδο. Η παράμετρος `model` είναι ένα μοντέλο Keras για κατηγοριοποίηση. Η `batchROIs` είναι ένας πίνακας NumPy που περιέχει ένα σύνολο (batch) περιοχών ενδιαφέροντος. Η `batchLocs` είναι ένα σύνολο συντεταγμένων (x,y) για κάθε περιοχή ενδιαφέροντος (ROI) που ανήκει στο `batchROI`. Η παράμετρος `labels` είναι ένα λεξικό ετικετών που διατηρείται καθόλη την διάρκεια της κατηγοριοποίησης. Το κλειδί του είναι το όνομα της ετικέτας ενώ οι τιμή είναι μια λίστα από πλειάδες. Η κάθε πλειάδα περιέχει τις συντεταγμένες των κουτιών που βρίσκονται γύρω από τα οχήματα και τις αντίστοιχες πιθανότητες των ετικετών/κλάσεων κατηγορίας τους. Μια άλλη παράμετρος της συνάρτησης είναι η `minProb`. Η προαναφερθείσα παράμετρος δηλώνει την ελάχιστη απαιτούμενη πιθανότητα που χρειάζεται μια κατηγοριοποίηση για να θεωρείται έγκυρη ανίχνευση. Συγκεκριμένα, χρησιμοποιούμε την παράμετρο αυτή για να φιλτράρουμε τις αδύνατες ανιχνεύσεις `k`. Η

επόμενη παράμετρος ονομάζεται `top` και αποθηκεύει το πλήθος των καλύτερων k προβλέψεων. Ακόμη, η παράμετρος `dims` αποθηκεύει τις χωρικές διαστάσεις του κουτιού (`bounding box`) που πρέπει να είναι οι ίδιες με τις χωρικές διαστάσεις του δικτύου.

Η έξοδος του τελευταίου επιπέδου συγκέντρωσης, που είναι επίπεδο συγκέντρωσης `max pooling` δύο διαστάσεων, είναι σχήματος $(N, 512, 7, 7)$, υποδηλώνοντας πως υπάρχουν 512 φίλτρα μεγέθους 7×7 . Για να χειριστούμε αυτές τις τιμές ως διάνυσμα χαρακτηριστικών χρειάζεται να τις `flatten` σε ένα πίνακα δύο διαστάσεων μεγέθους $(N, 25088)$, χρησιμοποιώντας την συνάρτηση `reshape` στα χαρακτηριστικά. Ακολούθως, πραγματοποιούμε τις προβλέψεις των πιθανοτήτων, (με την χρήση του `Logistic Regression`), καλώντας την συνάρτηση `predict` στο μοντέλο κατηγοριοποίησης με το `batch` των περιοχών ενδιαφέροντος (`batchROIs`). Όπως έχουμε αναφέρει σε προηγούμενη παράγραφο τα δίκτυα μας είναι προεκπαιδευμένα με τα βάρη `ImageNet` για αυτό πρέπει να αποκωδικοποιήσουμε τις προβλέψεις τους.

Στη συνέχεια, περνούμε πάνω από τις αποκωδικοποιημένες προβλέψεις των συνελκτικών δικτύων. Ακολούθως, φιλτράρουμε τις αδύνατες προβλέψεις διασφαλίζοντας πως η προβλεπόμενη πιθανότητα είναι μεγαλύτερη από την ελάχιστη απαιτούμενη πιθανότητα. Έπειτα, εξάγουμε τις συντεταγμένες των κουτιών για κάθε πιθανότητα για την οποία ισχύει η πιο πάνω συνθήκη. Παράλληλα, σχεδιάζουμε τα κουτιά γύρω από τα όχημα που έχουν προβλεφθεί. Μετά, ενημερώνουμε το λεξικό ετικετών, παίρνοντας μια λίστα προβλέψεων για την ετικέτα, προσθέτουμε το κουτί και την αντίστοιχη πιθανότητα στην λίστα. Αργότερα, επιστρέφουμε το λεξικό ετικετών στην συνάρτηση που το ζήτησε. Θα δούμε σε επόμενη ενότητα πως μπορούμε να συνδυάσουμε τις τεχνικές και τα αντίστοιχα κομμάτια κώδικα που δημιουργήσαμε για να κατασκευάσουμε ένα ανιχνευτή οχημάτων.

Επαναλαμβάνουμε την πιο πάνω διαδικασία για τις υπόλοιπες αρχιτεκτονικές δικτύων με μόνη διαφορά το σχήμα των χαρακτηριστικών π.χ. `MobileNet 224x224 pixels` $(N, 1024, 7, 7)$, `MobileNet 192x192 pixels` $(N, 1024, 6, 6)$,

MobileNet 160x160 pixels (N, 1024,5,5), MobileNet 128x128 pixels (N, 1024,4,4), κπλ.

5.10.4 Υλοποίηση Non maxima Suppression

Όπως είπαμε σε προηγούμενο κεφάλαιο, η παρουσία επικαλυπτόμενων κουτιών κατά την ανίχνευση των οχημάτων θα συμβεί μιας και είναι ένα σημάδι πως ο ανιχνευτής οχημάτων μαθαίνει να ανιχνεύει τα αντικείμενα που θέλουμε. Είναι χειρότερο ο ανιχνευτής να προβλέπει ένα αντικείμενο που δεν είναι όχημα ως τέτοιο (false positive) ή να μην ανιχνεύει ένα καθόλου ένα όχημα.

Ο λόγος που ανιχνεύονται πολλαπλά κουτιά για ένα όχημα οφείλεται στην χρήση του κινητού παραθύρου. Ένα κινητό παράθυρο μετακινείται από αριστερά προς δεξιά και από πάνω προς τα κάτω μέσα σε μια εικόνα, κατηγοριοποιώντας μια περιοχή ενδιαφέροντος εισόδου που λαμβάνει ως είσοδο σε κάθε βήμα. Επομένως, ένα κινητό παράθυρο κοινοποιεί ψηλότερη πιθανότητα για μια συγκεκριμένη ετικέτα όσο πιο κοντά φτάνει στο αντίστοιχο αντικείμενο. Μπορούμε να παρομοιάσουμε αυτή την λειτουργία του κινητού παραθύρου ανίχνευσης αντικειμένων με το γνωστό παιδικό παιχνίδι ζεστό/κρύο. Στο παιχνίδι όσο πιο κοντά περπατούσαμε στο κρυμμένο αντικείμενο τόσο πιο ζεστή ήταν η θέση μας ενώ στο παράδειγμα μας όσο πιο κοντά μετακινείται το παράθυρο στο αντικείμενο ενδιαφέροντος (όχημα) τόσο πιο ζεστή γίνεται η τοποθεσία στην οποία βρίσκεται και επομένως ανιχνεύει πιο πολλά κουτιά. Όσο πιο μακριά περπατούσαμε από το κρυμμένο αντικείμενο τόσο πιο κρύα γινόταν η θέση μας. Αντίστοιχα, όσο πιο μακριά μετακινείται το παράθυρο από το αντικείμενο ενδιαφέροντος, τόσο πιο κρύα γίνεται η θέση μας και ως εκ τούτου ανιχνεύει λιγότερα κουτιά.

Όταν συνδυάσουμε το κινητό παράθυρο με της πυραμίδες εικόνων, το δίκτυο εντοπίζει πολλά κουτιά που περικλείουν το αντικείμενο και σε διαφορετικές κλίμακες. Επομένως, χρειαζόμαστε ένα τρόπο με τον οποίο θα αγνοούμε τα πολλαπλά κουτιά εκτός αυτό με την μεγαλύτερη εμπιστοσύνη/μεγαλύτερη πιθανότητα.

Για να αφαιρέσουμε τα πολλαπλά κουτιά γύρω από το ίδιο αντικείμενο θα χρησιμοποιήσουμε το non maxima suppression. Το non maxima suppression είναι ένα φίλτρο που αφαιρεί τα επικαλυπτόμενα κουτιά. Αυτό, υπολογίζει την αναλογία επικάλυψης μεταξύ των κουτιών του ίδιου αντικειμένου και αφαιρεί αυτά που επικαλύπτονται πολύ μεταξύ τους.

Υλοποίηση Κυρίως Εφαρμογής Ανιχνευτών Οχημάτων:

Έχουμε ήδη περιγράψει πως υλοποιήσαμε διάφορες βοηθητικές συναρτήσεις που χρησιμοποιήσαμε. Σε αυτό το σημείο, θα περιγράψουμε την συνένωση τους ώστε αυτά να απαρτίζουν έναν ολοκληρωμένο ανιχνευτή οχημάτων. Πρώτα, δημιουργούμε ένα καινούργιο αρχείο με το όνομα `simple_detection_rc.py`. Έπειτα, φορτώνουμε τις κατάλληλες βιβλιοθήκες λογισμικού όπως την πυραμίδα εικόνων, το κινητό παράθυρο που δημιουργήσαμε μαζί σε ξεχωριστό αρχείο. Φορτώνουμε την συνάρτηση non maxima suppression από την βιβλιοθήκη `imutils`. Κατασκευάζουμε ένα επεξεργαστή των παραμέτρων που δίνονται στην γραμμή εντολών. Συγκεκριμένα οι παράμετροι που λαμβάνει η εφαρμογή από την γραμμή εντολών είναι η εικόνα εισόδου και η ελάχιστη πιθανότητα μια ανίχνευση να θεωρείται επιτυχής σε μια περιοχή ενδιαφέροντος (ROI). Η πιθανότητα αυτή, όπως αναφέραμε και σε προηγούμενη παράγραφο χρησιμοποιείται στο φιλτράρισμα των πολλαπλών αδύναμων ανιχνεύσεων του ίδιου αντικειμένου.

Στην συνέχεια, θα επεξηγήσουμε και θα αρχικοποιήσουμε σημαντικές παραμέτρους των ανιχνευτών. (Κατά την διεξαγωγή Πειραμάτων στους Ανιχνευτές Οχημάτων, θα τις μεταβάλλουμε για βρούμε τις κατάλληλες τιμές όσον αφορά το σύνολο δεδομένων μας και το πρόβλημα που θέλουμε να επιλύσουμε.

Επεξήγηση Παραμέτρων των Ανιχνευτών Οχημάτων:

Input Size: Είναι το μέγεθος της εικόνας εισόδου όπως αυτό εκφράζεται στον δισδιάστατο χώρο γραφικής απεικόνισης.

Pyr_Scale / Pyramid Scale: Είναι ο κλιμακωτός παράγοντας σύμφωνα με τον οποίο μεταβάλλεται το μέγεθος της εικόνας σε κάθε επίπεδο της πυραμίδας [13].

Win Step – Window Step: Ο όρος window step δείχνει πόσα pixels αγνοούμε (πηδούμε) τόσο στον x όσο και στον y άξονα [13].

ROI Size: Είναι ο όρος που χρησιμοποιείται για να περιγράψει το μέγεθος της περιοχής ενδιαφέροντος που μας ενδιαφέρει στην εκάστοτε εφαρμογή. Το μέγεθος της περιοχής ενδιαφέροντος εκφράζεται με τιμές στον δισδιάστατο χώρο γραφικής απεικόνισης [13].

Batch Size: Ο όρος batch size αφορά το πλήθος των παραδειγμάτων εκπαίδευσης που χρησιμοποιούνται για να εκπαιδεύσουν το δίκτυο σε μια επανάληψη [13].

Παράδειγμα Αρχικοποίησης Παραμέτρων Ανιχνευτών Οχημάτων:

INPUT_SIZE = (350, 350)

PYR_SCALE = 1.5

WIN_STEP = 16

ROI_SIZE = (224, 224)

BATCH_SIZE = 64

Ακολούθως, αρχικοποιούμε το δίκτυο που χρησιμοποιείται ως ανιχνευτής οχημάτων (π.χ VGG16) με τα προεκπαιδευμένα βάρη (που έχουν εκπαιδευτεί στο σύνολο δεδομένων) ImageNet. Έπειτα, αρχικοποιούμε το λεξικό ετικετών. Ο ανιχνευτής/συνελικτικό δίκτυο, θα προβλέψει την αντιστοιχία μεταξύ κλειδιών/ετικετών και τιμών. Η τιμή σε αυτό το λεξικό θα είναι οι καρτεσιανές συντεταγμένες (x,y) - (προβλεπόμενη θέση του οχήματος). Αργότερα, θα εφαρμόσουμε το φίλτρο Non maxima suppression σε κάθε κουτί και ετικέτα. Στην συνέχεια, φορτώνουμε την εικόνα εισόδου μας και μεταβάλλουμε το μέγεθος της

ώστε αυτό να είναι ίσο με το `INPUT_SIZE`. Μετά, αρχικοποιούμε τα `batch ROIS` και τα `batches` των καρτεσιανών συντεταγμένων (x,y) .

Περνούμε πάνω από κάθε κλίμακα που έχει παραχθεί στην πυραμίδα εικόνων. Σε κάθε κλίμακα εφαρμόζουμε το κινητό παράθυρο για να εξάγουμε την περιοχή ενδιαφέροντος σε κάθε `WIN_STEP`. Προ επεξεργαζόμαστε την περιοχή ενδιαφέροντος όπως κάναμε στην κατηγοριοποίηση. Δημιουργούμε τα `batchROIs` και `batchLocs` που θα χρησιμοποιήσουμε στην συνάρτηση `classify_batch`. Αν το `batch` είναι γεμάτο, τότε εφαρμόζουμε την συνάρτηση `classify_batch function` για να κατηγοριοποιήσουμε τις περιοχές ενδιαφέροντος της εικόνας εισόδου. Εφόσον χρησιμοποιούμε το `batch processing`, είναι πιθανόν πως θα βρεθούμε σε μια κατάσταση όπου δεν υπάρχουν άλλα επίπεδα στην πυραμίδα εικόνων που χρειάζεται να ελεγχθούν με το κινητό παράθυρο αλλά υπάρχουν ακόμη εικόνες στο `batch`. Επομένως, ελέγχουμε αν υπάρχουν εικόνες στο `batch`, αν ισχύει τότε εφαρμόζουμε εκ νέου την συνάρτηση `classify_batch`.

Το λεξικό ετικετών περιέχει τις ετικέτες των κλάσεων και τις αντίστοιχες συντεταγμένες των οχημάτων που βρίσκονται στην εικόνα εισόδου. Περνούμε πάνω από τις ετικέτες με στόχο να βρούμε τα οχήματα που ανιχνεύτηκαν. Συγκεκριμένα, ψάχνουμε το κλειδί `k` στις ετικέτες, διατηρούμε όλα τα κουτιά και τις αντίστοιχες πιθανότητες για την τρέχουσα ετικέτα. Έπειτα, σχεδιάζουμε τα κουτιά μέσα στην εικόνα χωρίς εφαρμογή του φίλτρου `Non Maxima Supression`. Ακολούθως, εξάγουμε το κουτί (x,y) και τις αντίστοιχες πιθανότητες από το λεξικό ετικετών με ετικέτα `k` και εφαρμόζουμε το φίλτρο `non-maxima suppression` στην εικόνα εισόδου και την αποθηκεύουμε. Έτσι, μένουν μόνο οι πραγματικές ανιχνεύσεις οχημάτων. Ακολούθως, σχεδιάζουμε τις θέσεις των οχημάτων πάνω στην αρχική εικόνα εισόδου και την αποθηκεύουμε. Στην οθόνη, εμφανίζουμε το πλήθος των πετυχημένων ανιχνεύσεων, την πιθανότητα της κάθε ανίχνευσης να ανήκει στην κατηγορία `car` και τον χρόνο εκτέλεσης του ανιχνευτή οχημάτων.

Επαναλάβαμε την πιο πάνω διαδικασία στις διαφορετικές αρχιτεκτονικές δικτύων που χρησιμοποιήσαμε και στα διαφορετικά μεγέθη δεδομένων εισόδου που δοκιμάσαμε σε κάθε αρχιτεκτονική.

Κεφάλαιο 6

Πειράματα και Αποτελέσματα

6.1 Πειράματα

6.1.2 Εξαγωγή Χαρακτηριστικών Οχημάτων

6.1.1.1.ResNet50

6.1.2.2 VGG16

6.1.2.3 ShallowNet

6.1.2.4 MobileNet

6.1.2.5 VGG16 Transfer Learning)

6.1.2.6 MobileNet Transfer Learning

6.1.2 Εκπαίδευση, Έλεγχος και Επικύρωση Μοντέλων Αναγνώρισης και Ανίχνευσης Οχημάτων

6.1.2.1.ResNet50

6.1.2.7 VGG16

6.1.2.8 ShallowNet

6.1.2.9 MobileNet

6.1.2.10 VGG16 Transfer Learning

6.1.2.11 MobileNet Transfer Learning

6.1.3 Εκπαίδευση και Έλεγχος Ανιχνευτών Οχημάτων

6.2.2.7 ResNet50

6.2.2.8 VGG16

6.2.2.9 ShallowNet

6.2.2.10 MobileNet

6.2.2.11 VGG16 (Fine-tuned)

6.2.2.12 MobileNet (Fine-tuned)

6.2 Αποτελέσματα

6.2.1 Αποτελέσματα Εξαγωγής Χαρακτηριστικών Οχημάτων

6.2.1.1.ResNet50

6.2.2.13 VGG16

6.2.2.14 ShallowNet

6.2.2.15 MobileNet

6.2.2.16 VGG16 Transfer Learning

6.2.2.17 MobileNet Transfer Learning

6.3.1 Αποτελέσματα Ανιχνευτών Οχημάτων

6.3.1.1 Αποτελέσματα Simple Detection Drone ResNet50 Logistic Regression

6.3.1.2 Αποτελέσματα Simple Detection Drone VGG16 Logistic Regression

6.3.1.3 Αποτελέσματα Simple Detection Drone ShallowNet Logistic Regression

6.3.1.4 Αποτελέσματα Simple Detection Drone MobileNet Logistic Regression

6.3.1.5 Αποτελέσματα Simple Detection Drone VGG16 Transfer Learning Logistic

Regression

6.3.1.6 Αποτελέσματα Simple Detection Drone MobileNet Transfer Learning

Logistic Regression

6.1 Πειράματα

6.1.1 Εξαγωγή Χαρακτηριστικών Οχημάτων

Στην παρούσα διπλωματική εργασία χρησιμοποιήσαμε 3 διαφορετικά είδη transfer learning. Σε αυτή την υποενότητα, έχουμε πραγματοποιήσει ένα συγκεκριμένο είδος transfer learning, την εξαγωγή χαρακτηριστικών για οχήματα σε εναέριες λήψεις όσον αφορά όλες τις διαφορετικές αρχιτεκτονικές δικτύων (ResNet50, VGG16, ShallowNet, MobileNet, VGG16 Transfer Learning, MobileNet Transfer Learning) με διαφορετικές διαστάσεις εικόνων (224x224 pixels, 197x197 pixels, 192x192 pixels, 160x160 pixels, 128x128 pixels) που θα χρησιμοποιήσουμε στους κατηγοριοποιητές οχημάτων.

Στον εξαγωγέα χαρακτηριστικών δίνουμε ως είσοδο τις εικόνες του συνόλου δεδομένων μας και τον εκπαιδεύουμε χρησιμοποιώντας το ίδιο δίκτυο με το οποίο θα εκπαιδεύσουμε τον κάθε κατηγοριοποιητή οχημάτων. Οι αρχικές διαστάσεις των εικόνων του συνόλου δεδομένων εισόδου, στις οποίες πραγματοποιήθηκε εξαγωγή χαρακτηριστικών, είναι οι προκαθορισμένες (224x224 pixels). Ακολούθως, σύμφωνα με τα αποτελέσματα των κατηγοριοποιητών (π.χ Precision 1.00) και των ανιχνευτών ήταν καλά (όλα τα αυτοκίνητα της εικόνας – 1 κουτί ανά αυτοκίνητο), κρατούσαμε τις διαστάσεις των εικόνων εισόδου ως έχει (αν τα αποτελέσματα ήταν καλά) ή μειώναμε τις διαστάσεις τους πρώτα σε 192x192 (εκτός στο δίκτυο ResNet50 που οι ελάχιστες διαστάσεις δεδομένων εισόδου είναι 197x197 pixels και στο ShallowNet του οποίου οι διαστάσεις εικόνων ορίστηκαν από την αρχή ως 32x32 pixels. Στο ResNet50 το επόμενο μέγεθος εικόνων που δοκιμάστηκε ήταν 197x197 pixels, 160x160pixels, 128x128 pixels.

Ο εξαγωγέας χαρακτηριστικών χρησιμοποιεί τα προεκπαιδευμένα βάρη του ImageNet ως τα βάρη του. Όπως περιγράψαμε στο προηγούμενο κεφάλαιο, ο εξαγωγέας χαρακτηριστικών εξαγει/μαθαίνει τα χαρακτηριστικά που χρειάζεται μέσω των επιπέδων συνέλιξης που διαθέτει μεταβάλλοντας τα βάρη του ανάλογα.

6.2 Εκπαίδευση, Έλεγχος και Επαλήθευση Μοντέλων Αναγνώρισης και Ανίχνευσης Οχημάτων

Έχουμε χρησιμοποιήσει αρκετές αρχιτεκτονικές συνελκτικών δικτύων ως κατηγοριοποιητές. Πρώτα, ρυθμίσαμε τις διαστάσεις των εικόνων του συνόλου δεδομένων να είναι αρχικά 224x224 για κάθε δίκτυο που εκπαιδεύεται. Στη συνέχεια, μειώσαμε τις διαστάσεις τους με τον τρόπο που περιγράψαμε στην προηγούμενη υποενότητα. Ακολούθως, αν τα αποτελέσματα των κατηγοριοποιητών και των ανιχνευτών οχημάτων εξακολουθούσαν να μην είναι καλά, εκπαιδεύσαμε τα δίκτυα με διαφορετικούς αλγόριθμους ενημέρωσης συναπτικών βαρών ή με τους ίδιους αλγόριθμους αλλά με διαφορετικές παραμέτρους σε αυτούς.

Στα δίκτυα VGG16 Transfer Learning, MobileNet Transfer Learning χρησιμοποιήσαμε τον αλγόριθμο ενημέρωσης συναπτικών βαρών RMSprop κατά την εκπαίδευση του τελευταίου πλήρως συνδεδεμένου τμήματος των δικτύων και ακολούθως τον SGD στην εκπαίδευση ολόκληρου του δικτύου πλην μερικών επιπέδων που κρατούσαμε παγωμένα. Συνήθως, στις αρχιτεκτονικές δικτύων που διαθέτουν πολλά επίπεδα, κρατούσαμε παγωμένα μόνο 15 επίπεδα από τα 90+ επίπεδα των συγκεκριμένων αρχιτεκτονικών.

Στα δίκτυα ResNet50, VGG16, ShallowNet, και MobileNet ο αλγόριθμος εκπαίδευσης των δικτύων ήταν ο Logistic Regression. Η είσοδος των δικτύων ήταν τα χαρακτηριστικά οχημάτων που εξάχθηκαν στην προηγούμενη υποενότητα.

Η επαλήθευση των μοντέλων/κατηγοριοποιητών πραγματοποιείται με το πέρασμα του συνόλου δεδομένων επαλήθευσης στο δίκτυο και το δίκτυο όντας εκπαιδευμένο επιλέγει πως η εικόνα περιέχει ένα όχημα (έξοδος δικτύου: 1) ή ότι δεν περιέχει (έξοδος δικτύου: 0) και αντιστοιχούμε την έξοδο του δικτύου με την κατάλληλη ετικέτα.

6.2.1 Εκπαίδευση, Έλεγχος και Επαλήθευση Μοντέλου Κατηγοριοποίησης (Κατηγοριοποιητή) ResNet50

ResNet50							
Experiment Number	Class	Input Size	Hyperparameter C	Precision	Recall	F1 Score	Support
1	car	(224, 224)	0.0001	0.98	1.00	0.99	206
1	No-car	(224, 224)	0.0001	1.00	0.96	0.98	169

Πίνακας 6.1

6.2.2 Εκπαίδευση, Έλεγχος και Επαλήθευση Μοντέλου Κατηγοριοποίησης (Κατηγοριοποιητή) VGG16

VGG16							
Experiment Number	Class	Input Size	Hyperparameter C	Precision	Recall	F1 Score	Support
1	car	(224, 224)	0.0001	1.00	1.00	1.00	184
1	No-car	(224, 224)	0.0001	1.00	1.00	1.00	191

Πίνακας 6.2

6.2.3 Εκπαίδευση, Έλεγχος και Επαλήθευση Μοντέλου Κατηγοριοποίησης (Κατηγοριοποιητή) ShallowNet

ShallowNet											
Experiment Number	Class	Input Size	Optimizer (lr, Nesterov, momentum)	Precision	Recall	F1 Score	Support	Acc	Loss	Val_Acc	Val_Loss
1	car	(32, 32)	SGD(lr=0.001)	0.84	0.96	0.90	169	0.91	0.31	0.90	0.31
1	No-car	(32, 32)	SGD(lr=0.001)	0.97	0.85	0.90	206	0.91	0.31	0.90	0.31
2	car	(32, 32)	SGD(lr=0.005)	0.79	0.99	0.88	169	0.93	0.18	0.87	0.26
2	No-car	(32, 32)	SGD(lr=0.005)	0.99	0.78	0.88	206	0.93	0.18	0.87	0.26
3	car	(32, 32)	SGD(lr=0.009)	0.84	0.99	0.91	169	0.90	0.18	0.95	0.19
3	No-car	(32, 32)	SGD(lr=0.009)	0.99	0.85	0.92	206	0.99	0.04	0.98	0.12
4	car	(32, 32)	SGD(lr=1e-3, momentum=1.0, nesterov=True)	0.99	0.99	0.99	169	0.99	0.04	0.98	0.12

4	No-car	(32, 32)	SGD(lr=1e-3, momentum=1.0, nesterov=True)	0.99	0.99	0.99	206	0.99	0.04	0.98	0.12
5	car	(32, 32)	SGD(lr=1e-3, decay=1e-6, momentum=0.9, nesterov=True)	0.98	1.00	0.99	169	0.99	0.03	0.98	0.14
5	No-car	(32, 32)	SGD(lr=1e-3, decay=1e-6, momentum=0.9, nesterov=True)	1.00	0.98	0.99	206	0.99	0.03	0.98	0.14

Πίνακας 6.3

6.2.4 Εκπαίδευση, Έλεγχος και Επαλήθευση Μοντέλου Κατηγοριοποίησης (Κατηγοριοποιητή) MobileNet

MobileNet							
Experiment Number	Class	Input Size	Hyperparameter C	Precision	Recall	F1 Score	Support
1	car	(224, 224)	0.0001	1.00	1.00	1.00	177
1	No-car	(224, 224)	0.0001	1.00	1.00	1.00	173
2	car	(192, 192)	0.01	1.00	1.00	1.00	174
2	No-car	(192, 192)	0.01	1.00	1.00	1.00	201
3	No-car	(128, 128)	0.001	1.00	1.00	1.00	174
3	No-car	(128, 128)	0.001	1.00	1.00	1.00	226

Πίνακας 6.4

6.2.5 Εκπαίδευση, Έλεγχος και Επαλήθευση Μοντέλου Κατηγοριοποίησης (Κατηγοριοποιητή) VGG16 Transfer Learning

VGG16 Transfer Learning												
Experiment Number	Before/After Transfer Learning	Class	Input Size	Optimizer (lr, Nesterov, momentum)	Precision	Recall	F1 Score	Support	Acc	Loss	Val_Acc	Val_Loss

1	before	car	(192, 192)	RMSprop(lr=0.001)	1.00	1.00	1.00	175	0.99	0.00002	0.98	0.004
1	before	No-car	(192, 192)	RMSprop(lr=0.001)	1.00	1.00	1.00	200	0.99	0.00002	0.98	0.004
1	after	car	(192, 192)	SGD(lr=0.001)	1.00	1.00	1.00	175	0.99	0.00002	0.98	0.004
1	after	No-car	(192, 192)	SGD(lr=0.001)	1.00	1.00	1.00	200	0.99	0.00002	0.98	0.004

Πίνακας 6.5

6.2.6 Εκπαίδευση, Έλεγχος και Επαλήθευση Μοντέλου Κατηγοριοποίησης (Κατηγοριοποιητή) MobileNet Transfer Learning

MobileNet Transfer Learning												
Experiment Number	Before /After Transfer Learning	Class	Input Size	Optimizer (lr, Nesterov, momentum)	Precision	Recall	F1 Score	Support	Acc	Loss	Val_Acc	Val_Loss
1	before	car	(224, 224)	RMSprop(lr=0.0001)	1.00	0.62	0.77	206	0.99	0.09	0.79	2.3
1	before	No-car	(224, 224)	RMSprop(lr=0.0001)	0.68	1.00	0.81	169	0.99	0.09	0.79	2.3
1	after	car	(224, 224)	SGD(lr=0.001)	0.97	1.00	0.99	206	1.00	0.03	0.98	1.1921e-07
1	after	No-car	(224, 224)	SGD(lr=0.001)	1.00	0.96	0.98	169	1.00	0.03	0.98	1.1921e-07
2	before	car	(192, 192)	RMSprop(lr=0.0001)	1.00	1.00	1.00	206	1.00	3.4443e-06	0.74	0.33
2	before	No-car	(192, 192)	RMSprop(lr=0.0001)	0.99	1.00	1.00	169	1.00	3.4443e-06	0.74	0.33
2	after	car	(192, 192)	SGD(lr=0.001)	0.99	1.00	0.99	206	1.00	3.2323e-04	0.99	0.03
2	after	No-car	(192, 192)	SGD(lr=0.001)	1.00	0.98	0.99	169	1.00	3.2323e-04	0.99	0.03
4	before	car	(160, 160)	RMSprop(lr=0.0001)	1.00	0.51	0.68	206	0.99	0.09	0.79	2.37
4	before	No-car	(160, 160)	RMSprop(lr=0.0001)	0.63	1.00	0.77	169	0.99	0.09	0.79	2.37
4	after	car	(160, 160)	SGD(lr=0.001)	1.00	0.99	0.99	206	1.00	3.6182e-04	1.00	2.5943e-04
4	after	No-car	(160, 160)	SGD(lr=0.001)	0.98	1.00	0.99	169	1.00	3.6182e-04	1.00	2.5943e-04

Πίνακας 6.6

6.3 Μετρικές

Το πρόβλημα που επιλύουμε είναι να εντοπίσουμε αν υπάρχει ένα τουλάχιστον όχημα μέσα σε μια εικόνα που λήφθηκε από drone, την ακριβή πρόβλεψη της θέσης του οχήματος ή των οχημάτων στην εικόνα. Ως εκ τούτου, χρησιμοποιήθηκαν εικόνες που απεικόνιζαν διαφορετικούς τύπους οχημάτων και εικόνες που δεν περιείχαν οχήματα κατά την εκπαίδευση και την επαλήθευση του δικτύου [9].

Τα δεδομένα εισόδου είναι μια εικόνα που περιέχει ή δεν περιέχει οχήματα με μια συγκεκριμένη μορφή. Τα δεδομένα εξόδου είναι η προβλεπόμενη εικόνα στην οποία το δίκτυο έχει εντοπίσει αν υπάρχει τουλάχιστον ένα όχημα ή καθόλου οχήματα και οι θέσεις (συντεταγμένες των εντοπισθέντων οχημάτων με μια συγκεκριμένη μορφή που είναι παρόμοια με την μορφή των δεδομένων εισόδου [9].

Η μέτρηση της ακρίβειας των μοντέλων που εκπαιδεύτηκαν και επαληθεύτηκαν έγινε με την χρήση διάφορων μετρικών που χρησιμοποιούνται συχνά σε προβλήματα μηχανικής μάθησης.

Ακολούθως, παρουσιάζουμε τις μετρικές μέσω των οποίων έχουμε την δυνατότητα να συγκρίνουμε την απόδοση διαφορετικών κατηγοριοποιητών και ανιχνευτών [9].

Η συνολική ακρίβεια (Accuracy)- το πλήθος των συνολικών σωστών προβλέψεων είναι σημαντικό μέτρο απόδοσης και ορίζεται από το πηλίκο των σωστών προβλέψεων προς το συνολικό πλήθος των προβλέψεων

$$Accuracy = \frac{TP + TN}{FP + FN + TP + TN}$$

Εξίσωση 6.1 – Συνολική Ακρίβεια

Ο ρυθμός σφάλματος είναι επίσης μια χρήσιμη μετρική με την οποία μπορούμε να μετρήσουμε την απόδοση ενός μοντέλου. Η μετρική αυτή ορίζεται από το πηλίκο των λανθασμένων προβλέψεων προς το συνολικό πλήθος των προβλέψεων. Ο ρυθμός σφάλματος μπορεί να οριστεί και ως το αντίστροφο της ακρίβειας – των συνολικών σωστών προβλέψεων [9].

$$Error\ Rate = 1 - Accuracy = \frac{FP + FN}{FP + FN + TP + TN}$$

Εξίσωση 6.2 – Ρυθμός Σφάλματος

Ακρίβεια – Precision είναι το πλήθος των ορθών προβλέψεων που διαιρείται με το άθροισμα των σωστών προβλέψεων με αυτό των λανθασμένων θετικών προβλέψεων. Η προαναφερθείσα τιμή ονομάζεται προβλεπτική τιμή σωστής κατηγοριοποίησης (Positive Predictive Value – PPV) και μας λέει το ποσοστό των αντικειμένων που εντόπισε σωστά ο κατηγοριοποιητής (ποσοστό πραγματικά σωστών αντικειμένων). Όταν η ακρίβεια έχει χαμηλή τιμή τότε μπορούμε να καταλήξουμε στο συμπέρασμα πως υπάρχει μεγάλο πλήθος false positive (λανθασμένων θετικών προβλέψεων) [9].

$$Precision = \frac{TP}{TP + FP}$$

Εξίσωση 6.3 – Ακρίβεια

Πλήθος σωστών απαντήσεων (Recall) είναι το πλήθος των σωστών προβλέψεων που διαιρείται με το άθροισμα των σωστών προβλέψεων με αυτό τον λανθασμένων προβλέψεων. Ονομάζεται και ρυθμός θετικών προβλέψεων (true positive rate) ή ευαισθησία (sensitivity). Με την χρήση της μετρικής της ανάκλησης προτύπων (Recall), λέμε πως το μοντέλο εντόπισε ένα συγκεκριμένο ποσοστό των αντικειμένων του συνόλου δεδομένων. Όταν το πλήθος των σωστών απαντήσεων έχει χαμηλή τιμή τότε αυτό μας οδηγεί στο συμπέρασμα πως υπάρχει μεγάλος αριθμός false negative προβλέψεων (λανθασμένων αρνητικών προβλέψεων) [9].

$$Recall = \frac{TP}{TP + FN}$$

Εξίσωση 6.4 – Recall

Συνήθως, χρησιμοποιείται ο δείκτης F1 που είναι ο αρμονικός αριθμητικός μέσος των μετρικών Precision και Recall καθώς οι μετρικές Precision και Recall συμπληρώνουν η μια την άλλη [9].

$$F_1 = \frac{2(Precision \cdot Recall)}{Precision + Recall}$$

Εξίσωση 6.5 – F1

6.4 Εκπαίδευση και Έλεγχος Ανιχνευτών Οχημάτων

Input Image:



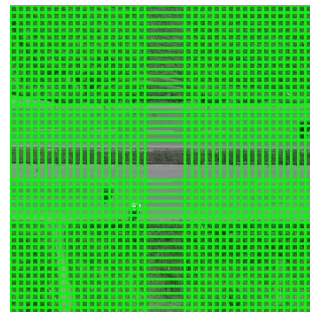
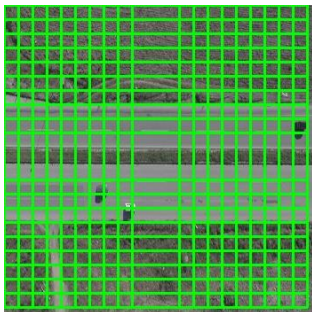
Σχήμα 6.1 Εικόνα Εισόδου των Πειραμάτων που πραγματοποιήθηκαν στους Ανιχνευτές Οχημάτων

6.4.1 Simple Detection Drone ResNet Logistic Regression

	Simple Detection Drone ResNet50 Logistic Regression
--	---

Experiment Number	Variables:				
	INPUT_SIZ E	PYR_SCALE	WIN_STEP	ROI_SIZE	BATCH_SIZ E
1	(350, 350)	1.5	16	(197, 197)	128
2	(350, 350)	1.5	8	(197, 197)	128

Πίνακας 6.7



Εικόνες 6.2, 6.3 Simple Detection Drone ResNet50
Logistic Regression Experiment 1

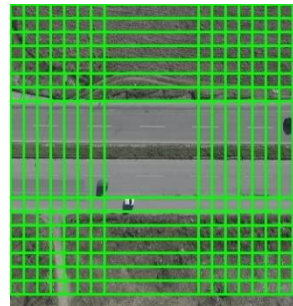
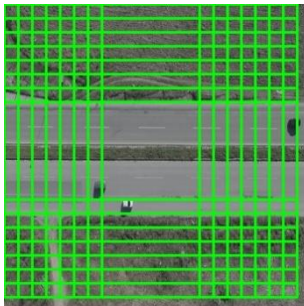
Εικόνες 6.4, 6.5 Simple Detection Drone ResNet50
Logistic Regression Experiment 2

6.4.2 Simple Detection Drone VGG16 Logistic Regression

Simple Detection Drone VGG16 Logistic Regression					
Experiment Number	Variables:				
	INPUT SIZE	PYR SCALE	WIN STEP	ROI SIZE	BATCH SIZE
8	(350, 350)	1.5	16	(224, 224)	256
9	(350, 350)	1.25	16	(224, 224)	256
10	(350, 350)	1.05	16	(224, 224)	256
12	(350, 350)	1.15	16	(224, 224)	256

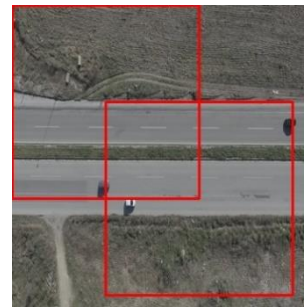
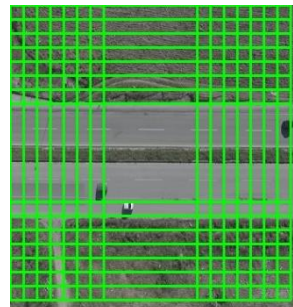
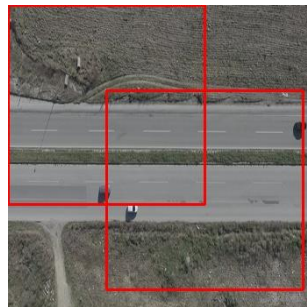
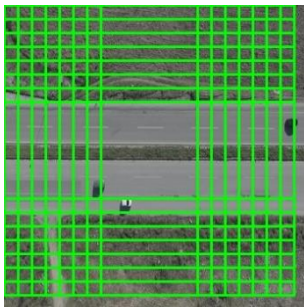
13	(350, 350)	1.01	16	(224, 224)	256
14	(350, 350)	1.01	4	(224, 224)	256
29	(350, 350)	1.22	8	(224, 224)	128
30	(350, 350)	1.22	4	(224, 224)	128
31	(350, 350)	1.22	2	(224, 224)	128
41	(350, 350)	1.5	16	(192, 192)	256
42	(350, 350)	1.5	8	(192, 192)	256
43	(350, 350)	1.5	4	(192, 192)	256

Πίνακας 6.8



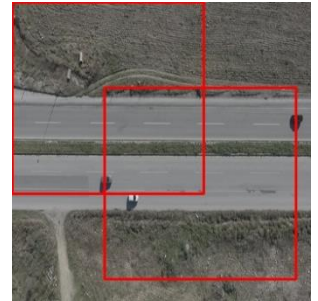
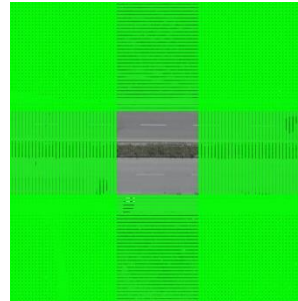
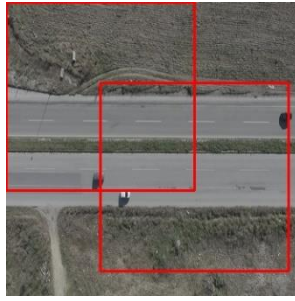
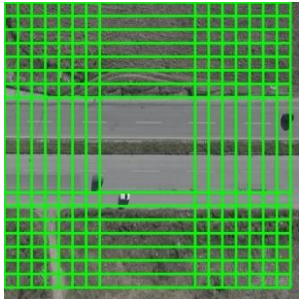
Εικόνες 6.6, 6.7 Simple Detection Drone VGG16 Logistic Regression Experiment 8

Εικόνες 6.8, 6.9 Simple Detection Drone VGG16 Logistic Regression Experiment 9



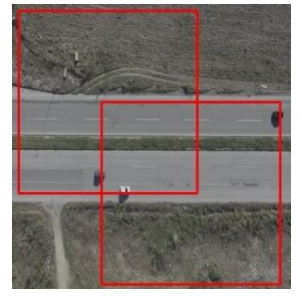
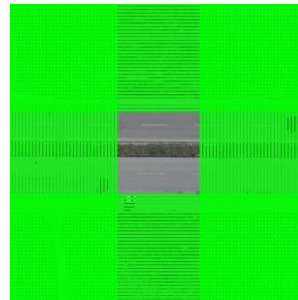
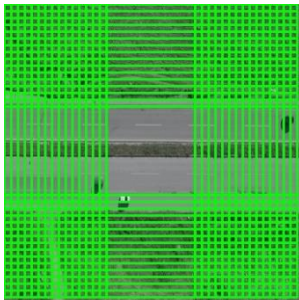
Εικόνες 6.10, 6.11 Simple Detection Drone VGG16 - Logistic Regression Experiment 10

Εικόνες 6.12, 6.13 Simple Detection Drone VGG16 - Logistic Regression Experiment 12



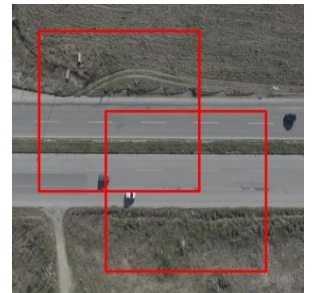
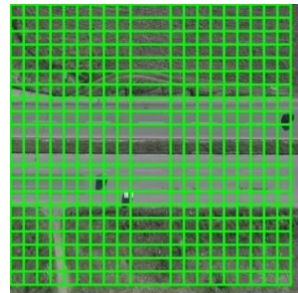
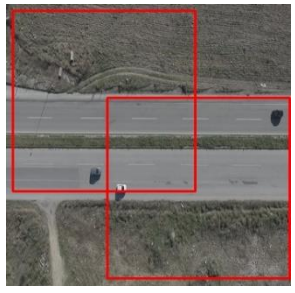
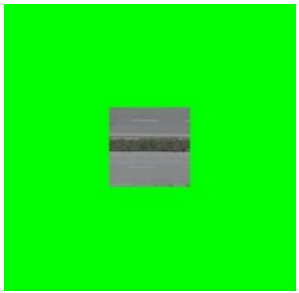
**Εικόνες 6.14, 6.15 Simple Detection Drone
VGG16 - Logistic Regression Experiment 13**

**Εικόνες 6.16, 6.17 Simple Detection Drone
VGG16 - Logistic Regression Experiment 14**



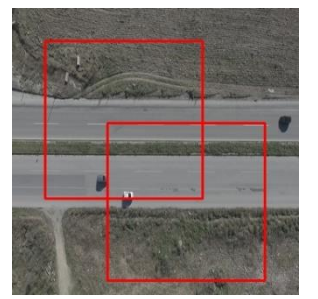
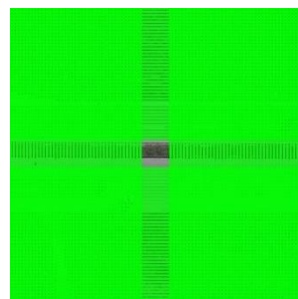
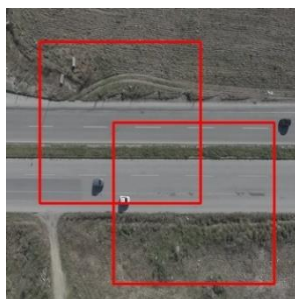
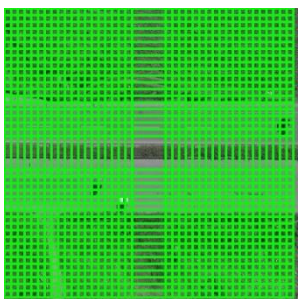
**Εικόνες 6.18, 6.19 Simple Detection Drone
VGG16 - Logistic Regression Experiment 29**

**Εικόνες 6.20, 6.21 Simple Detection Drone
VGG16 - Logistic Regression Experiment 30**



**Εικόνες 6.22, 6.23 Simple Detection Drone
VGG16 - Logistic Regression Experiment 31**

**Εικόνες 6.24, 6.25 Simple Detection Drone
VGG16 - Logistic Regression Experiment 41**



Εικόνες 6.26, 6.27 Simple Detection Drone
VGG16 - Logistic Regression Experiment 42

Εικόνες 6.28, 6.29 Simple Detection Drone
VGG16 - Logistic Regression Experiment 43

6.4.3 Simple Detection Drone ShallowNet Logistic Regression

Simple Detection Drone ShallowNet Logistic Regression					
Experiment Number	Variables:				
	INPUT SIZE	PYR SCALE	WIN STEP	ROI SIZE	BATCH SIZE
12	(350, 350)	1.25	16	(32, 32)	256
18	(350, 350)	1.14	16	(32, 32)	256
20	(350, 350)	1.06	16	(32, 32)	256
22	(350, 350)	1.26	16	(32, 32)	256

Πίνακας 6.9



Εικόνες 6.30, 6.31 Simple Detection Drone
ShallowNet Logistic Regression Experiment 12

Εικόνες 6.32, 6.33 Simple Detection Drone
ShallowNet Logistic Regression Experiment 18



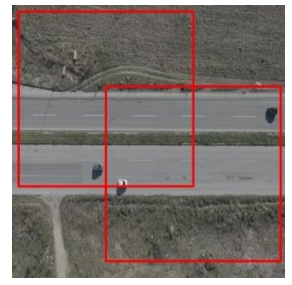
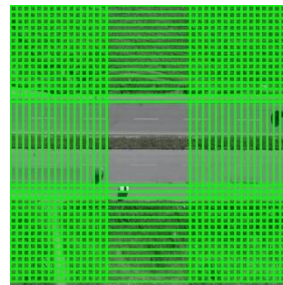
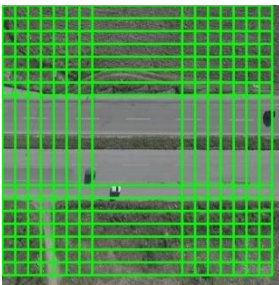
Εικόνες 6.34, 6.35 Simple Detection Drone

Εικόνες 6.36, 6.37 Simple Detection Drone

6.4.4 Simple Detection Drone MobileNet Logistic Regression

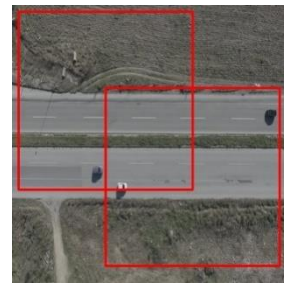
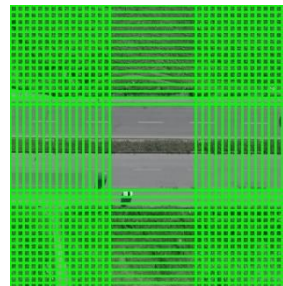
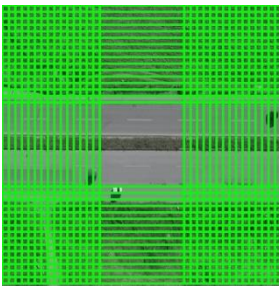
Simple Detection Drone MobileNet Logistic Regression					
Experiment Number	Variables:				
	INPUT SIZE	PYR SCALE	WIN STEP	ROI SIZE	BATCH SIZE
1	(350, 350)	1.5	16	(224, 224)	128
2	(350, 350)	1.5	8	(224, 224)	128
4	(350, 350)	1.25	8	(224, 224)	128
3	(350, 350)	1.5	8	(224, 224)	64
7	(350, 350)	1.22	8	(224, 224)	64
8	(350, 350)	1.21	8	(224, 224)	64

Πίνακας 6.10

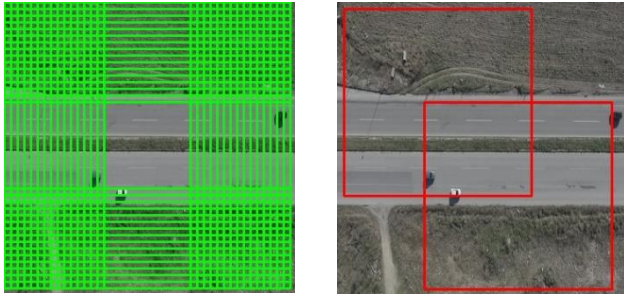


Εικόνες 6.38, 6.39 Detection Drone MobileNet Logistic Regression Experiment 1

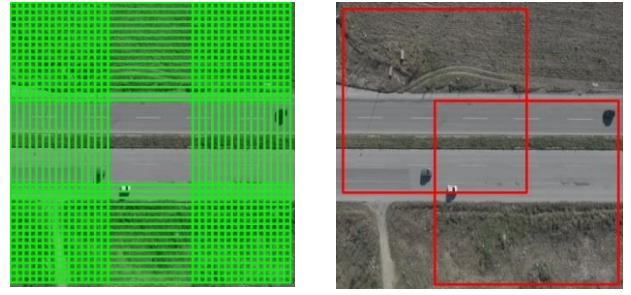
Εικόνες 6.40, 6.41 Detection Drone MobileNet Logistic Regression Experiment 2



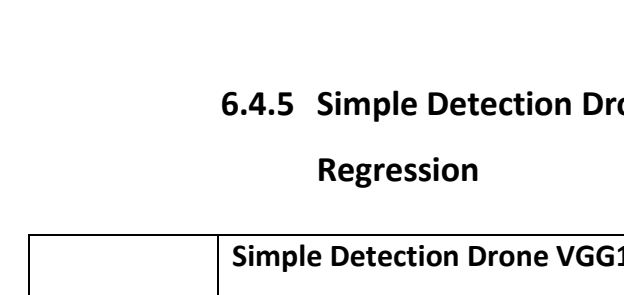
Εικόνες 6.42, 6.43 Detection Drone MobileNet Logistic Regression Experiment 4



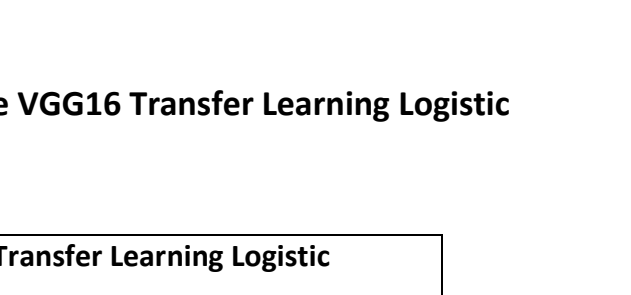
Εικόνες 6.44, 6.45 Detection Drone MobileNet Logistic Regression Experiment 3



Εικόνες 6.46, 6.47 Detection Drone MobileNet Logistic Regression Experiment 7



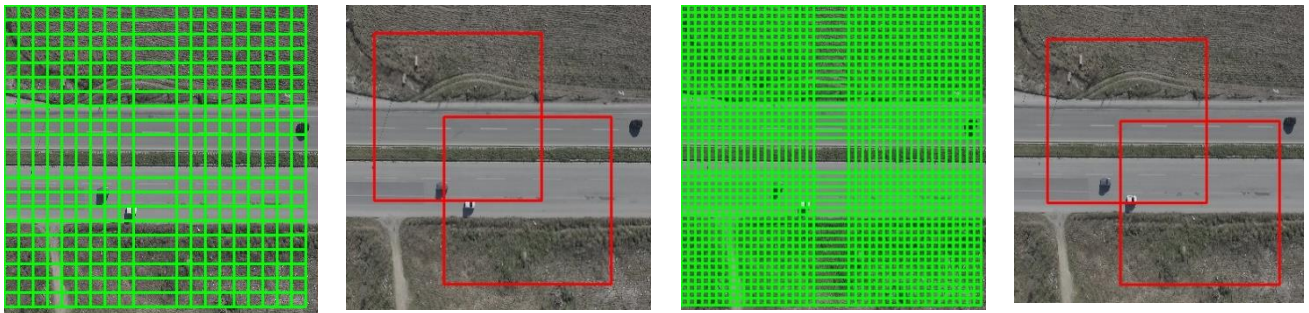
Εικόνες 6.48, 6.49 Detection Drone MobileNet Logistic Regression Experiment 8



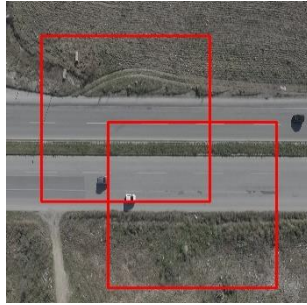
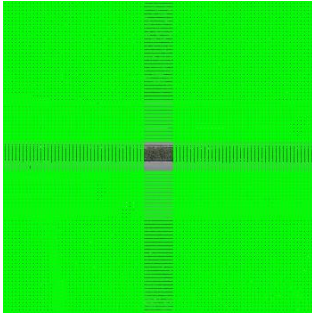
6.4.5 Simple Detection Drone VGG16 Transfer Learning Logistic Regression

Simple Detection Drone VGG16 Transfer Learning Logistic Regression					
Experiment Number	Variables:				
	INPUT SIZE	PYR SCALE	WIN STEP	ROI SIZE	BATCH SIZE
1	(350, 350)	1.5	16	(192, 192)	64
2	(350, 350)	1.5	8	(192, 192)	64
3	(350, 350)	1.5	4	(192, 192)	64

Πίνακας 6.11



**Εικόνες 6.50, 6.51 Simple Detection Drone
VGG16 Transfer Learning Logistic Regression
Experiment 1**



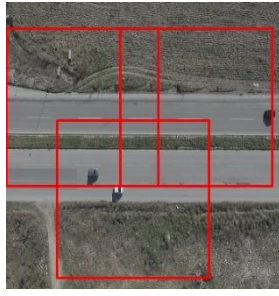
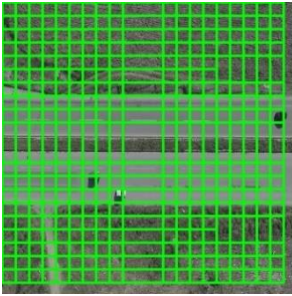
**Εικόνες 6.52, 6.53 Simple Detection Drone
VGG16 Transfer Learning Logistic Regression
Experiment 2**

**Εικόνες 6.54, 6.55 Simple Detection Drone
VGG16 Transfer Learning Logistic Regression
Experiment 3**

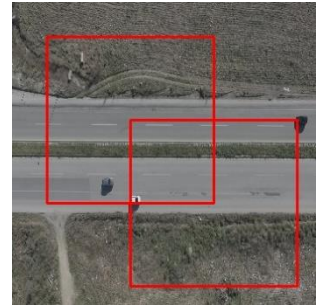
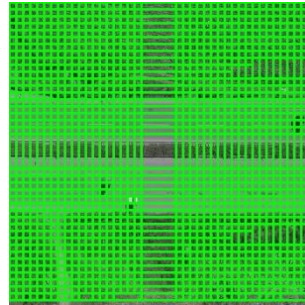
6.4.6 Simple Detection Drone MobileNet Transfer Learning Logistic Regression

Simple Detection Drone MobileNet Transfer Learning Logistic Regression					
Experiment Number	Variables:				
	INPUT SIZE	PYR SCALE	WIN STEP	ROI SIZE	BATCH SIZE
1	(350, 350)	1.5	16	(192, 192)	64
2	(350, 350)	1.5	8	(192,192)	64
3	(350, 350)	1.5	4	(192,192)	64

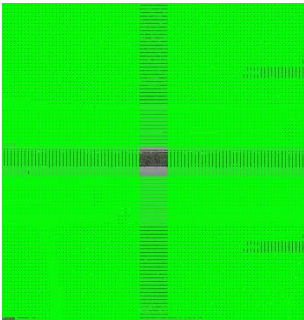
Πίνακας 6.12



**Εικόνες 6.56, 6.57 Simple Detection Drone
MobileNet Transfer Learning Logistic Regression
Experiment 1**



**Εικόνες 6.58, 6.59 Simple Detection Drone
MobileNet Transfer Learning Logistic
Regression Experiment 2**



**Εικόνες 6.60, 6.61 Simple Detection Drone
MobileNet Transfer Learning Logistic Regression
Experiment 3**

6.5 Αποτελέσματα

6.5.1 Αποτελέσματα Εξαγωγής Χαρακτηριστικών Οχημάτων

Τα χαρακτηριστικά των οχημάτων εξάχθηκαν με επιτυχία. Η εξαγωγή χαρακτηριστικών πραγματοποιήθηκε μόνο στις εξής εκδόσεις των δικτύων (ResNet50, VGG16, ShallowNet, MobileNet) και για κάθε διαφορετική διάσταση εικόνων που χρησιμοποιήθηκε ως είσοδος σε αυτά.

Στην εικόνα 6.62, μπορούμε να δούμε πως φαίνεται η επιτυχής εκτέλεση και ολοκλήρωση της διαδικασίας εξαγωγής χαρακτηριστικών οχημάτων χρησιμοποιώντας μια συγκεκριμένη αρχιτεκτονική δικτύου. Παρατηρούμε πως στην γραμμή εντολών βλέπουμε

εκτός από την πρόοδο του εξαγωγέα χαρακτηριστικών και τον χρόνο που διήρκησε η εξαγωγή τους.

```

(16, 2048)
(16, 2048)
Extracting Features: 91% |##### | ETA: 0:00:17(
16, 224, 224, 3)
(16, 2048)
(16, 2048)
Extracting Features: 92% |##### | ETA: 0:00:15(
16, 224, 224, 3)
(16, 2048)
(16, 2048)
Extracting Features: 93% |##### | ETA: 0:00:13(
16, 224, 224, 3)
(16, 2048)
(16, 2048)
Extracting Features: 94% |##### | ETA: 0:00:10(
16, 224, 224, 3)
(16, 2048)
(16, 2048)
Extracting Features: 96% |##### | ETA: 0:00:08(
16, 224, 224, 3)
(16, 2048)
(16, 2048)
Extracting Features: 97% |##### | ETA: 0:00:06(
16, 224, 224, 3)
(16, 2048)
(16, 2048)
Extracting Features: 98% |##### | ETA: 0:00:03(
12, 224, 224, 3)
(12, 2048)
(12, 2048)
Extracting Features: 100% |##### | Time: 0:03:29
(d14cv) pyimagesearch@pyimagesearch-d14cv:~/d14cv/PB_Code/chapter10-dogs_vs_cats$
  
```

Εικόνες 6.62, Εξαγωγή Χαρακτηριστικών – Όψη Αποτελεσμάτων

6.5.2 Αποτελέσματα Μοντέλων Κατηγοριοποίησης/Κατηγοριοποιητών Οχημάτων

Parameters											
Experiment Number	Before /After Transfer Learning	Class	Network	Input Size	Hyperparameter C	Optimizer (lr, Nesterov, momentum)	Precision	Recall	F1 Score	Support	time
2		car	ResNet50	(197, 197)	0.001	/	0.98	1.00	0.98	175	40 sec
2		No-car	ResNet50	(197, 197)	0.001	/	1.00	0.99	1.00	200	40 sec
2		car	VGG16	(192, 192)	0.0001	/	1.00	1.00	1.00	175	45 sec
2		No-car	VGG16	(192, 192)	0.0001	/	1.00	1.00	1.00	200	45 sec
6		car	ShallowNet	(32, 32)	/	Nadam(lr=1e-3)	1.00	1.00	1.00	206	45 sec
6		No-car	ShallowNet	(32, 32)	/	Nadam(lr=1e-3)	1.00	1.00	1.00	169	45 sec
4		No-car	MobileNet	(160, 160)	0.0001	/	1.00	1.00	1.00	181	40 sec
4		No-car	MobileNet	(160, 160)	0.0001	/	1.00	1.00	1.00	194	40 sec

2	before	car	VGG16 Transfer Learning	(192, 192)	/	RMSprop(lr=0.001)	1.00	1.00	1.00	175	3.28 h mins
2	before	No-car	VGG16 Transfer Learning	(192, 192)	/	RMSprop(lr=0.001)	1.00	1.00	1.00	200	3.28 h mins
2	after	car	VGG16 Transfer Learning	(192, 192)	/	SGD(lr=0.001)	1.00	1.00	1.00	175	3.28 h mins
2	after	No-car	VGG16 Transfer Learning	(192, 192)	/	SGD(lr=0.001)	1.00	1.00	1.00	200	3.28 h mins
3	before	car	MobileNet Transfer Learning	(192, 192)	/	RMSprop(lr=0.001)	1.00	1.00	1.00	206	2.38 h mins
3	before	No-car	MobileNet Transfer Learning	(192, 192)	/	RMSprop(lr=0.001)	1.00	1.00	1.00	169	2.38 h mins
3	after	car	MobileNet Transfer Learning	(192, 192)	/	SGD(lr=0.001)	1.00	1.00	1.00	206	2.38 h mins
3	after	No-car	MobileNet Transfer Learning	(192, 192)	/	SGD(lr=0.001)	1.00	1.00	1.00	169	2.38 h mins

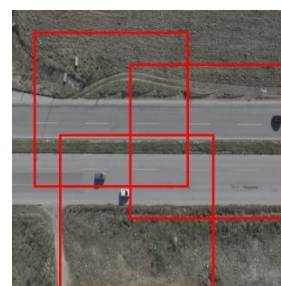
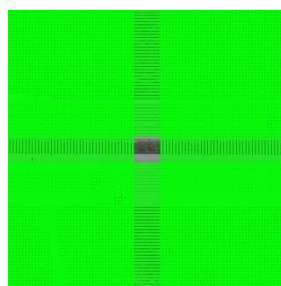
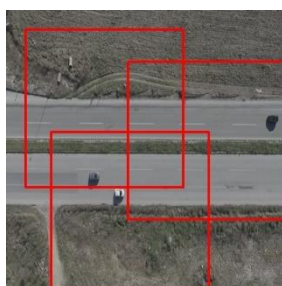
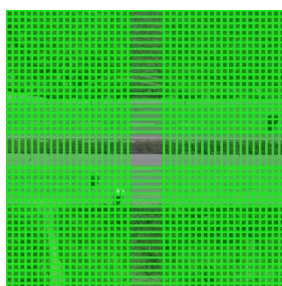
6.5.3 Αποτελέσματα Ανιχνευτών Οχημάτων

Το καλύτερα πειράματα των ανιχνευτών οχημάτων, που εκπαιδεύτηκαν επίσης με τον αλγόριθμο μάθησης Logistic Regression, βρίσκονται πιο κάτω. Στον πίνακα που παρουσιάζουμε, βλέπουμε τις παραμέτρους που χρησιμοποιήθηκαν για να παραχθούν τα καλύτερα πειράματα.

Οι τιμές που βρίσκονται στον πιο κάτω πίνακα αφορούν την εικόνα εισόδου που δώσαμε στα δίκτυα.

		OBJECT DETECTION RESULTS	
Experiment Number	Network	Parameters	

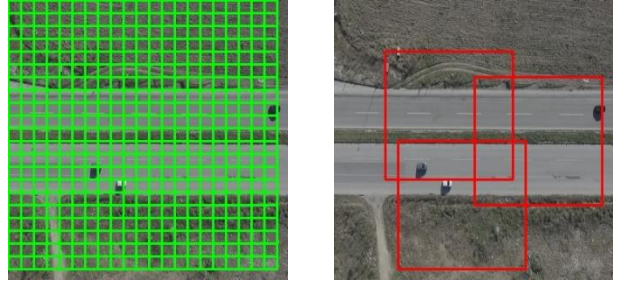
		INPUT _SIZE	PYR_S CALE	WIN_ STEP	ROI_SIZE	BATCH _SIZE	Result	time
3	ResNet50	(350, 350)	1.25	8	(197, 197)	128	3/3 3 bounding boxes	135 sec
45	VGG16	(350, 350)	1.25	4	(192, 192)	64	3/3 3 bounding boxes	117 sec
52	ShallowNet	(350, 350)	1.5	4	(32, 32)	64	1/3 1 bounding box	3 sec
10	MobileNet	(350, 350)	1.5	16	(160, 160)	64	3/3 3 bounding boxes	30 sec
4	VGG16 Transfer Learning	(350, 350)	1.25	4	(192, 192)	64	3/3 3 bounding boxes	111 sec
4	MobileNet Transfer Learning	(350, 350)	1.25	4	(192, 192)	64	3/3 3 bounding boxes	26.97 sec



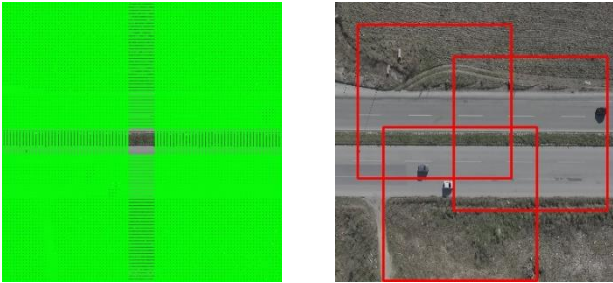
**Εικόνες 6.63, 6.64 Αποτελέσματα Simple
Detection Drone ResNet50 Logistic Regression**



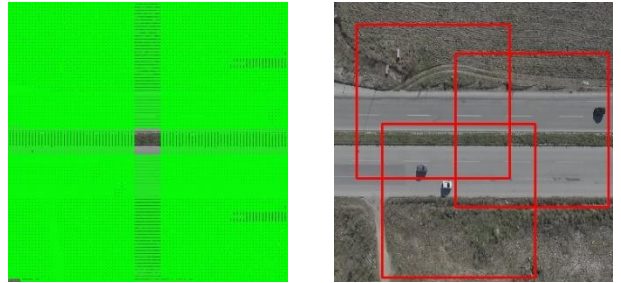
**Εικόνες 6.65, 6.66 Αποτελέσματα Simple
Detection Drone VGG16 Logistic Regression**



**Εικόνες 6.67, 6.68 Αποτελέσματα Simple
Detection Drone ShallowNet Logistic Regression**



**Εικόνες 6.69, 6.70 Αποτελέσματα Simple
Detection Drone MobileNet Logistic
Regression**



**Εικόνες 6.71, 6.72 Αποτελέσματα Simple
Detection Drone VGG16 Transfer Learning
Logistic Regression**



**Εικόνες 6.73, 6.74 Αποτελέσματα Simple
Detection Drone MobileNet Transfer Learning
(Fine-tuned) Logistic Regression**



Κεφάλαιο 7

Συμπεράσματα

7.1 Συμπεράσματα	1
7.2 Μελλοντική Εργασία	3

7.1 Συμπεράσματα

Τα αποτελέσματα τόσο των κατηγοριοποιητών αντικειμένων όσο και των ανιχνευτών ήταν εξαιρετικά (ποσοστά ακριβείας εκπαίδευσης 99-100%, ποσοστά ακριβείας επαλήθευσης 98-100% . Οι ανιχνευτές που συνιστώ να χρησιμοποιηθούν σε σχεδόν πραγματικού χρόνου εφαρμογές ανίχνευσης και αναγνώρισης οχημάτων είναι ο MobileNet Transfer Learning και ακολούθως ο MobileNet, καθώς είναι οι δύο πιο γρήγοροι ανιχνευτές οχημάτων (26 sec, 30 sec αντίστοιχα). Οι γενικοί χρόνοι εκτέλεσης τους που λήφθηκαν κατά την διάρκεια των πειραμάτων, ήταν από 1 - 26.97 sec και 2 – 30 sec. Τα άνω όρια των χρόνων εκτέλεσης λήφθηκαν από τα καλύτερα πειράματα/αποτελέσματα. Τα 6/6 δίκτυα που χρησιμοποιήθηκαν ως κατηγοριοποιητές οχημάτων δίνουν άριστα αποτελέσματα όπως, αναφέραμε και πιο πάνω, Από την άλλη, οι 5/6 ανιχνευτές αντικειμένων βρίσκουν όλα τα αυτοκίνητα που υπάρχουν μέσα σε μια εικόνα, (στο παράδειγμά μας 3/3 με ένα κουτί γύρω από κάθε αυτοκίνητο).

Στον ανιχνευτή οχημάτων ShallowNet με Logistic Regression υπάρχει περιθώριο βελτίωσης περιθώρια βελτίωσης και στον ανιχνευτή οχημάτων της συγκεκριμένης αρχιτεκτονικής μαζί με τον συγκεκριμένο αλγόριθμο καθώς προβλεπόταν ορθά η ύπαρξη και η θέση 1/3 αυτοκίνητα της εικόνας. Ο λόγος για τον οποίο τα ποσοστά ακριβείας είναι πιο χαμηλά σε σχέση με τα άλλα δίκτυα είναι πως το ShallowNet διαθέτει πολύ λιγότερα επίπεδα συνέλιξης και επομένως, ο αντίστοιχος κατηγοριοποιητής, δεν προλαβαίνει να μάθει καλά τα ακριβή χαρακτηριστικά ενός οχήματος και να παράξει ένα ακριβές μοντέλο.

Οι χρόνοι εκτέλεσης των κατηγοριοποιητών που λήφθηκαν κατά των καλύτερων πειραμάτων ήταν από 30 - 45 sec όσον αφορά τα δίκτυα ResNet50, VGG16, ShallowNet, MobileNet και 2μιση – 3μιση ώρες στα δίκτυα VGG16 Transfer Learning & MobileNet Transfer Learning.

Οι χρόνοι εκτέλεσης των ανιχνευτών ResNet50, VGG16, VGG16 finetuned, ShallowNet, που λήφθηκαν κατά την διάρκεια των πειραμάτων ήταν από 1 sec– 3 hours. Τα άνω όρια των χρόνων εκτέλεσης λήφθηκαν από τα καλύτερα πειράματα/αποτελέσματα.

Επίσης, παρατηρήθηκε πως όσο πιο μικρές είναι οι διαστάσεις της εικόνας εισόδου τόσο πιο πολύ μειώνονται τα ποσοστά ακριβείας εκπαίδευσης και επαλήθευσης (-3% για κάθε μείωση των διαστάσεων (π.χ. MobileNet 224x224 - ποσοστά ακριβείας εκπαίδευσης και επαλήθευσης 99% και 100% αντίστοιχα τότε στο MobileNet 160x160 τα ποσοστά αυτά θα γίνουν 93% και 94%).

Ακόμη, στους ανιχνευτές οχημάτων δοκιμάστηκαν πολύ ψηλές τιμές στην παράμετρο batch size (1024, 2048, 4096) αλλά παρατηρήθηκε πως τα δίκτυα δεν συνέκλιναν ποτέ ή συνέκλιναν μετά από 3 – 5 μέρες. Επομένως, συστήνουμε την χρήση των τιμών 64, 128, 256 στην παράμετρο batch size καθώς τα δίκτυα συγκλίνουν πιο γρήγορα σε τοπικό ελάχιστο με αυτές τις τιμές και η ακρίβεια εντοπισμού ενός οχήματος παραμένει εξαιρετική.

Επιπρόσθετα, χρησιμοποιήθηκαν αρκετοί αλγόριθμοι ενημέρωσης συναπτικών βαρών π.χ. RMSprop, SGD μόνο με ρυθμό μάθησης ή με ρυθμό μάθησης και ορμή ή με ρυθμό μάθησης ορμή και επιτάχυνση Nesteron, Nadam, Adam. Έγινε εμφανές πως ο αλγόριθμος Adam παρέχει τα καλύτερα ποσοστά ακριβείας εκπαίδευσης και ειδικότερα σε δίκτυα που έχουν λίγα επίπεδα στην αρχιτεκτονική τους π.χ. ShallowNet. Επιπλέον, παρατηρήθηκε ότι ο αλγόριθμος Nadam δίνει τα καλύτερα ποσοστά ακριβείας επαλήθευσης και τα καλύτερα ποσοστά ακριβείας εκπαίδευσης σε όλες τις αρχιτεκτονικές και ειδικά σε αρχιτεκτονικές δικτύων που διαθέτουν λίγα επίπεδα. Ακόμη, οι αλγόριθμοι Adam και Nadam δίνουν τα καλύτερα ποσοστά σφάλματος εκπαίδευσης (δεν ισχύει το ίδιο για τα ποσοστά σφάλματος επαλήθευσης). Επίσης, αποδείχθηκε πως η χρήση των SGD μόνο με ρυθμό μάθησης, SGD με ρυθμό μάθησης και ορμή, SGD με ρυθμό μάθησης, ορμή και επιτάχυνση Nesteron δίνουν εξαιρετικά χαμηλά ποσοστά σφάλματος επαλήθευσης και ειδικότερα σε δίκτυα

που διαθέτουν πολλά επίπεδα. Στις που υλοποιήθηκαν με την χρήση εγχείρησης συνελκτικών δικτύων παρατηρήθηκε πως η χρήση του SGD με ρυθμό μάθησης μετά το Transfer Learning και συγκεκριμένα με χαμηλή τιμή ρυθμού μάθησης ήταν αρκετή για να μας δώσει ψηλά ποσοστά ακριβείας εκπαίδευσης και επαλήθευσης και χαμηλά ποσοστά σφάλματος εκπαίδευσης και επαλήθευσης.

Τέλος, Η χρήση του Transfer Learning κάνει εφικτή την υλοποίηση σχεδόν πραγματικού χρόνου αυτοματοποιημένων εφαρμογών στις οποίες οι μηχανές αναγνωρίζουν και εντοπίζουν με ακρίβεια διαφορετικά αντικείμενα όπως τα οχήματα και μπορούν να χρησιμοποιηθούν σε πολλές και χρήσιμες εργασίες οι οποίες εκτελούνται μέχρι στιγμής από συστήματα που δεν είναι τόσο ακριβή υπό την επιτήρηση ανθρώπων (π.χ. παρακολούθηση τροχαίας κίνησης) ή από ανθρώπους (π.χ. παρακολούθηση και εντοπισμός υπηρεσιακών οχημάτων – ταξί, λεωφορεία, ασθενοφόρα, περιπολικά κλπ) .

7.2 Μελλοντική Εργασία

Προτείνουμε να δοκιμαστούν και άλλες fine-tuned εκδόσεις προ εκπαιδευμένων δικτύων στην κατηγοριοποίηση και στην ανίχνευση οχημάτων (resNet50 fine-tuned, shallowNet fine-tuned) που δεν δοκιμάστηκαν στην παρούσα διπλωματική εργασία λόγω έλλειψης χρόνου. Επιπρόσθετα, εισηγούμαστε να γίνουν πειράματα με διαφορετικούς αλγορίθμους ενημέρωσης συναπτικών βαρών και διαφορετικές τιμές στις παραμέτρους τους (κατά την εκπαίδευση των κατηγοριοποιητών οχημάτων).

Ακόμη, υπάρχουν περιθώρια δοκιμών των εξ ολοκλήρου βαθιάς μάθησης ανιχνευτών αντικειμένων π.χ. Fast R-CNN, Mask RCNN, SSD, Yolo κ.ά. σε εικόνες για σχεδόν πραγματικού χρόνου ανιχνεύσεις οχημάτων. Επιπλέον, προτείνουμε να δοκιμαστούν τα πιο πάνω δίκτυα σε βίντεο με στόχο την ακριβή πραγματικού χρόνου ιχνηλάτηση οχημάτων. Η εφαρμογή που μόλις έχουμε περιγράψει μπορεί να χρησιμοποιηθεί σε

έξυπνες πόλεις και συγκεκριμένα στην αυτόματη παρακολούθηση της τροχαίας κίνησης, στην αυτόματη παρακολούθηση υπηρεσιακών οχημάτων (περιπολικών, ασθενοφόρων, πυροσβεστικών, ταξί, οχημάτων παράδοσης αγαθών) και στην αυτόματη παρακολούθηση των μέσων μαζικής μεταφοράς της πόλης (π.χ. λεωφορεία).

Οι δύο τρόποι ανίχνευσης και αναγνώρισης αντικειμένων που υπάρχουν μπορούν να χρησιμοποιηθούν σε ιατρικές εφαρμογές με στόχο την πρόβλεψη μιας μελλοντικής ασθένειας (πλάκα στην καρωτιδική αρτηρία, Parkinson, Alzheimer, κλπ.) σύμφωνα με ιατρικά δεδομένα (MRI, CT-SCAN) που λήφθηκαν πρόσφατα από τους ασθενείς.

Βιβλιογραφία

- [1] Ανθίμου, Μ., 2017, Deep Learning in Neural Networks (Presentation).
- [2] CS231 Website (University of Groningen), 2017, Convolutional Networks, <http://cs231n.github.io/convolutional-networks/#case>
- [3] Culfaz, F., 2018, Transfer Learning using MobileNet and Keras, <https://towardsdatascience.com/transfer-learning-using-mobilenet-and-keras-c75daf7ff299>
- [4] Das, S., 2017, CNN Architectures: Le Net, AlexNet, VGG, GoogLeNet, ResNet and more <https://medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5>
- [5] Διονυσίου, Α., 2018 Πρόβλεψη Δευτεροταγούς Δομής Πρωτεϊνών Με Χρήση Συνελικτικών Νευρωνικών Δικτύων σε Συνδυασμό Με Φίλτρα Gabor και Support Vector Machines.
- [6] Donges, N., 2018, Transfer Learning, <https://towardsdatascience.com/transfer-learning-946518f95666>
- [7] Jordan, J., Convnet Architectures, <https://www.jeremyjordan.me/convnet-architectures/>
- [8] Low, S., 2018, SqueezeNet and MobileNet: Deep learning models for mobile phones, <https://aiinpractice.com/squeezenet-mobilenet/>
- [9] Παππάς, Κ., 2018, Computer Vision for Aerial Applications.
- [10] Petkov, N., 2017, Deep Learning and Convolutional Neural Networks (Presentation).
- [11] Rosebrock., A., 2018, A gentle guide to Deep Learning Object Detection, <https://www.pyimagesearch.com/2018/05/14/a-gentle-guide-to-deep-learning-object-detection/>
- [12] Rosebrock, A., 2018, Deep Learning for Computer Vision with Python Starter Bundle (2nd Edition).
- [13] Rosebrock, A., 2018, Deep Learning for Computer Vision with Python Practitioner Bundle (2nd Edition).

- [14] Rosebrock., A., 2015, Image Pyramids with Python and OpenCV, <https://www.pyimagesearch.com/2015/03/16/image-pyramids-with-python-and-opencv/>
- [15] Rosebrock., A., 2015, Sliding Windows for Object Detection with Python and OpenCV, <https://www.pyimagesearch.com/2015/03/23/sliding-windows-for-object-detection-with-python-and-opencv/>
- [16] Χριστοδούλου, Χ., Διαφάνειες Μαθήματος ΕΠΛ442.
- [17] Wiggers., K., 2019, Believe it or not, Facebook is using AI to reduce abuse and bias, <https://venturebeat.com/2019/05/01/believe-it-or-not-facebook-is-using-ai-to-reduce-abuse-and-bias/>
- [18] Wikipedia, Machine Learning, https://en.wikipedia.org/wiki/Portal:Machine_learning
- [19] YouTube, a Eurovision song created by Artificial Intelligence: Blue Jeans and Bloody Tears, https://www.youtube.com/watch?v=4MKAf6YX_7M&fbclid=IwAR0niG56zCch6KjYEhve82R-unJ7s_sN0m_DL8x9JMB_I-NQk1S3YwtUKk

Παράρτημα Α - Εντολές Εκτέλεσης Εφαρμογών:

Πριν να εκτελέσουμε οποιοδήποτε κώδικα του Παραρτήματος Α, εκτελούμε την εντολή: `workon dl4cv` για να ενεργοποιήσουμε το περιβάλλον βαθιάς μηχανικής μάθησης.

Εντολές Εκτέλεσης Εξαγωγέων Χαρακτηριστικών:

Διεύθυνση Μονοπατιού Κώδικα: `/home/pyimagesearch/dl4cv/PB_Code/chapter10-dogs_vs_cats/`

Εξαγωγή Χαρακτηριστικών Δικτύου ResNet50:

Εικόνες 224x224 Pixels: `python extract_features_drone_pc.py --dataset /Data/drone_dataset/train_p_n/equal_size --output /Data/drone_dataset/train_p_n/output/features_ResNet_drone_equal_224_pc.hdf5`

Εικόνες 197x197 Pixels: `python extract_features_drone_pc.py --dataset /Data/drone_dataset/train_p_n/equal_size --output /Data/drone_dataset/train_p_n/output/features_ResNet_drone_equal_197_pc.hdf5`

Εξαγωγή Χαρακτηριστικών Δικτύου VGG16:

Εικόνες 224x224 Pixels: `python extract_features_drone_VGG16_pc.py --dataset /Data/drone_dataset/train_p_n/equal_size --output /Data/drone_dataset/train_p_n/output/features_VGG16_drone_equal_pc.hdf5`

Εικόνες 192x192 Pixels: `python extract_features_drone_VGG16_192_pc.py --dataset /Data/drone_dataset/train_p_n/equal_size --output /Data/drone_dataset/train_p_n/output/features_VGG16_drone_equal_192_pcpcpc.hdf5`

Εξαγωγή Χαρακτηριστικών Δικτύου MobileNet:

Εικόνες 224x224 Pixels: `python extract_features_drone_MobileNet_pc.py --dataset /Data/drone_dataset/train_p_n/equal_size --output /Data/drone_dataset/train_p_n/output/features_MobileNet_drone_equal_pcpc.hdf5`

Εικόνες 192x192 Pixels: `python extract_features_drone_MobileNet_2_pc.py --dataset /Data/drone_dataset/train_p_n/equal_size --output /Data/drone_dataset/train_p_n/output/features_MobileNet_drone_equal_192_pc.hdf5`

Εικόνες 160x160 Pixels: python extract_features_drone_MobileNet_4_pc.py --dataset /Data/drone_dataset/train_p_n/equal_size --output /Data/drone_dataset/train_p_n/output_pc/features_MobileNet_drone_equal_160_pc.hdf5

Εικόνες 128x128 Pixels: python extract_features_drone_MobileNet_3_pc.py --dataset /Data/drone_dataset/train_p_n/equal_size --output /Data/drone_dataset/train_p_n/output/features_MobileNet_drone_equal_128_pc.hdf5

Εντολές Εκτέλεσης Κατηγοριοποιητών Οχημάτων:

Διεύθυνση Μονοπατιού Κώδικα: /home/pyimagesearch/dl4cv/PB_Code/chapter10-dogs_vs_cats/

Κατηγοριοποιητής Οχημάτων ResNet50:

Εικόνες 224x224 Pixels: python train_model_drone_pc.py --db /Data/drone_dataset/train_p_n/output/features_ResNet_drone_equal_224_pc.hdf5 --model /Data/drone_dataset/train_p_n/output/resnet50_pc.pickle

Εικόνες 197x197 Pixels: python train_model_drone_ResNet50_197_pc.py --db /Data/drone_dataset/train_p_n/output_pc/features_ResNet_drone_equal_197_pcpcpcpc.hdf5 --model /Data/drone_dataset/train_p_n/output/drone_ResNet50_equal_197_test_pc.pickle

Κατηγοριοποιητής Οχημάτων VGG16:

Εικόνες 224x224 Pixels: python train_model_drone_VGG16_pc.py --db /Data/drone_dataset/train_p_n/output/features_VGG16_drone_equal_pc.hdf5 --model /Data/drone_dataset/train_p_n/output/drone_equal.pickle

Εικόνες 192x192 Pixels: python train_model_drone_VGG16_pc.py --db /Data/drone_dataset/train_p_n/output/features_VGG16_drone_equal_192_pcpcpc.hdf5 --model /Data/drone_dataset/train_p_n/output/drone_equal_VGG16_192_pc.pickle

Κατηγοριοποιητής Οχημάτων ShallowNet:

Διεύθυνση Μονοπατιού Κώδικα: /home/pyimagesearch/dl4cv/SB_Code/chapter13-saving_loading

Εικόνες 32x32 Pixels: python shallownet_train_drone_pc.py --dataset /Data/drone_dataset/train_p_n/equal_size --model /Data/drone_dataset/train_p_n/output/shallownet_weights_drone_32_02_test_pc.hdf5

Κατηγοριοποιητής Οχημάτων MobileNet:

Εικόνες 224x224 Pixels: python train_model_drone_MobileNet_pc.py --db /Data/drone_dataset/train_p_n/output/features_MobileNet_drone_equal_pc_2.hdf5 --model /Data/drone_dataset/train_p_n/output/drone_equal_MobileNet_pc.pickle

Εικόνες 192x192 Pixels: python train_model_drone_MobileNet_2_pc.py --db /Data/drone_dataset/train_p_n/output_pc/features_MobileNet_drone_equal_192_pc.hdf5 --model /Data/drone_dataset/train_p_n/output/drone_equal_MobileNet_192_pc.pickle

Εικόνες 160x160 Pixels: python train_model_drone_MobileNet_4_pc.py --db /Data/drone_dataset/train_p_n/output_pc/features_MobileNet_drone_equal_160_pc.hdf5 --model /Data/drone_dataset/train_p_n/output_pc/drone_equal_MobileNet_160_pc.pickle

Εικόνες 128x128 Pixels: python train_model_drone_MobileNet_3_pc.py --db /Data/drone_dataset/train_p_n/output/features_MobileNet_drone_equal_128_pc.hdf5 --model /Data/drone_dataset/train_p_n/output/drone_equal_MobileNet_128_pc.pickle

Κατηγοριοποιητής Οχημάτων VGG16 Transfer Learning:

Εικόνες 192x192 Pixels: python finetune_VGG16_pc.py --dataset /Data/drone_dataset/train_p_n/equal_size --model /Data/drone_dataset/train_p_n/output_pc/finetune_VGG16_192_pc.model

Κατηγοριοποιητής Οχημάτων MobileNet Transfer Learning:

Εικόνες 192x192 Pixels: python finetune_MobileNet_pc.py --dataset /Data/drone_dataset/train_p_n/equal_size --model /Data/drone_dataset/train_p_n/output_pc/finetune_MobileNet_pc_192.model

Εικόνες 160x160 Pixels: python finetune_MobileNet_2_pc.py --dataset /Data/drone_dataset/train_p_n/equal_size --model /Data/drone_dataset/train_p_n/output_pc/finetune_MobileNet_pc_160.model

Εντολές Εκτέλεσης Ανιχνευτών Οχημάτων:

Διεύθυνση Μονοπατιού Κώδικα: /home/pyimagesearch/dl4cv/PB_Code/chapter13-fund_obj_det

Ανιχνευτής Οχημάτων ResNet50:

Εικόνες 224x224 Pixels: python simple_detection.py --image 0001.png --confidence 0.9

Εικόνες 197x197 Pixels: python sd_drone_pc_RsN_LgRegg.py --image 0001.png --confidence 0.9

Ανιχνευτής Οχημάτων VGG16:

Εικόνες 224x224 Pixels: python simple_detection_drone_VGG16_pc_2.py --image 0001.png --confidence 0.9

Εικόνες 192x192 Pixels: python simple_detection_drone_VGG16_pc.py --image 0001.png --confidence 0.9

Ανιχνευτής Οχημάτων ShallowNet:

Εικόνες 32x32 Pixels: python simple_detection_drone_shallow_net_pc.py --image 0001.png --confidence 0.9

Ανιχνευτής Οχημάτων MobileNet:

Εικόνες 192x192 Pixels: : python simple_detection_drone_mobileNet_pc.py --image 0001.png --confidence 0.9

Ανιχνευτής Οχημάτων VGG16 Transfer Learning:

Εικόνες 224x224 Pixels: python simple_detection_fine-tuning_VGG16_pc_2.py --image 0001.jpg --confidence 0.9 --model /Data/drone_dataset/train_p_n/output_pc/VGG16_drone_pc.model

Εικόνες 192x192 Pixels: python simple_detection_fine-tuning_VGG16_pc.py --image 0001.jpg --confidence 0.9 --model /Data/drone_dataset/train_p_n/output_pc/finetune_VGG16_192_pc.model

Ανιχνευτής Οχημάτων MobileNet Transfer Learning:

Εικόνες 192x192 Pixels: python simple_detection_fine-tuning_mobileNet_pc.py --image 0001.jpg --confidence 0.9 --model /Data/drone_dataset/train_p_n/output_pc/finetune_MobileNet_pc_192.model

```
pyimagesearch@pyimagesearch-d14cv: ~/dl4cv/P8_Code/chapter10-dogs_vs_cats
Epoch 5/25
35/35 [=====] - 77s 2s/step - loss: 0.0382 - acc: 0.987
5 - val_loss: 0.0111 - val_acc: 1.0000
Epoch 6/25
35/35 [=====] - 80s 2s/step - loss: 0.0180 - acc: 0.995
5 - val_loss: 0.0078 - val_acc: 1.0000
Epoch 7/25
35/35 [=====] - 83s 2s/step - loss: 0.0235 - acc: 0.988
4 - val_loss: 0.0408 - val_acc: 1.0000
Epoch 8/25
35/35 [=====] - 101s 3s/step - loss: 0.0107 - acc: 0.99
73 - val_loss: 0.0051 - val_acc: 0.9973
Epoch 9/25
35/35 [=====] - 136s 4s/step - loss: 0.0178 - acc: 0.99
20 - val_loss: 0.0060 - val_acc: 1.0000
Epoch 10/25
35/35 [=====] - 104s 3s/step - loss: 0.0105 - acc: 0.99
55 - val_loss: 0.0051 - val_acc: 0.9973
Epoch 11/25
35/35 [=====] - 95s 3s/step - loss: 0.0266 - acc: 0.991
7 - val_loss: 0.0039 - val_acc: 1.0000
Epoch 12/25
35/35 [=====] - 94s 3s/step - loss: 0.0128 - acc: 0.995
5 - val_loss: 0.0032 - val_acc: 1.0000
Epoch 13/25
35/35 [=====] - 93s 3s/step - loss: 0.0090 - acc: 0.996
4 - val_loss: 0.0030 - val_acc: 1.0000
Epoch 14/25
35/35 [=====] - 93s 3s/step - loss: 0.0095 - acc: 0.996
4 - val_loss: 0.0050 - val_acc: 0.9973
Epoch 15/25
34/35 [=====>.] - ETA: 1s - loss: 0.0143 - acc: 0.9926
```

Σχήμα A.1 – Εκπαίδευση και Επαλήθευση Μοντέλου Κατηγοριοποίησης (Transfer Learning)

```
pyimagesearch@pyimagesearch-d14cv: ~/d14cv/SB_Code/chapter13-saving_loading
Epoch 7/20
1125/1125 [-----] - 1s 606us/step - loss: 0.3718 - acc: 0.8391 - val_loss: 0.3202 - val_acc: 0.8800
Epoch 8/20
1125/1125 [-----] - 1s 609us/step - loss: 0.3830 - acc: 0.8516 - val_loss: 0.3489 - val_acc: 0.8373
Epoch 9/20
1125/1125 [-----] - 1s 614us/step - loss: 0.3210 - acc: 0.8764 - val_loss: 0.2765 - val_acc: 0.9120
Epoch 10/20
1125/1125 [-----] - 1s 599us/step - loss: 0.2992 - acc: 0.8942 - val_loss: 0.3976 - val_acc: 0.8133
Epoch 11/20
1125/1125 [-----] - 1s 606us/step - loss: 0.3427 - acc: 0.8649 - val_loss: 0.2700 - val_acc: 0.9120
Epoch 12/20
1125/1125 [-----] - 1s 605us/step - loss: 0.3292 - acc: 0.8711 - val_loss: 0.4440 - val_acc: 0.7600
Epoch 13/20
1125/1125 [-----] - 1s 599us/step - loss: 0.2473 - acc: 0.9227 - val_loss: 0.5635 - val_acc: 0.7120
Epoch 14/20
1125/1125 [-----] - 1s 622us/step - loss: 0.3098 - acc: 0.8818 - val_loss: 0.6484 - val_acc: 0.7200
Epoch 15/20
1125/1125 [-----] - 1s 611us/step - loss: 0.2650 - acc: 0.9084 - val_loss: 0.2195 - val_acc: 0.9307
Epoch 16/20
1125/1125 [-----] - 1s 588us/step - loss: 0.2356 - acc: 0.9236 - val_loss: 0.3118 - val_acc: 0.8453
Epoch 17/20
1125/1125 [-----] - 1s 580us/step - loss: 0.2418 - acc: 0.9156 - val_loss: 0.7406 - val_acc: 0.6987
Epoch 18/20
1125/1125 [-----] - 1s 602us/step - loss: 0.2342 - acc: 0.9138 - val_loss: 0.2505 - val_acc: 0.8987
Epoch 19/20
1125/1125 [-----] - 1s 593us/step - loss: 0.2212 - acc: 0.9333 - val_loss: 0.1993 - val_acc: 0.9280
Epoch 20/20
1125/1125 [-----] - 1s 595us/step - loss: 0.2558 - acc: 0.9022 - val_loss: 0.1972 - val_acc: 0.9280
[INFO] serializing network...
[INFO] evaluating network...
      precision    recall  f1-score   support

 no-car    0.93    0.94    0.93     206
   car    0.93    0.91    0.92     169

 micro avg    0.93    0.93    0.93     375
 macro avg    0.93    0.93    0.93     375
weighted avg    0.93    0.93    0.93     375
```

Σχήμα A.2 – Εκπαίδευση και Επαλήθευση Μοντέλου Κατηγοριοποίησης (Transfer Learning)

```
pyimagesearch@pyimagesearch-d14cv: ~/d14cv/SB_Code/chapter13-saving_loading
Epoch 17/20
1125/1125 [-----] - 1s 588us/step - loss: 0.0693 - acc: 0.9840 - val_loss: 0.1310 - val_acc: 0.9813
Epoch 18/20
1125/1125 [-----] - 1s 585us/step - loss: 0.0211 - acc: 0.9956 - val_loss: 0.1575 - val_acc: 0.9813
Epoch 19/20
1125/1125 [-----] - 1s 602us/step - loss: 0.0175 - acc: 0.9947 - val_loss: 0.1144 - val_acc: 0.9893
Epoch 20/20
1125/1125 [-----] - 1s 593us/step - loss: 0.0448 - acc: 0.9911 - val_loss: 0.1290 - val_acc: 0.9893
[INFO] serializing network...
[INFO] evaluating network...
      precision    recall  f1-score   support

 no-car    0.99    0.99    0.99     206
   car    0.99    0.99    0.99     169

 micro avg    0.99    0.99    0.99     375
 macro avg    0.99    0.99    0.99     375
weighted avg    0.99    0.99    0.99     375

(d14cv) pyimagesearch@pyimagesearch-d14cv:~/d14cv/SB_Code/chapter13-saving_loading$ AC
(d14cv) pyimagesearch@pyimagesearch-d14cv:~/d14cv/SB_Code/chapter13-saving_loading$ python shallownet_train_drone_pc.py --dataset /Data/drone_dataset/train_p_n/equal_size --model /Data/drone_dataset/train_p_n/output/shallownet_weights_drone_32_02_test3_pc.hdf5
Using TensorFlow backend.
[INFO] loading images...
[INFO] processed 500/1500
[INFO] processed 1000/1500
[INFO] processed 1500/1500
[INFO] compiling model...
[INFO] training network...
Train on 1125 samples, validate on 375 samples
Epoch 1/20
2019-05-22 11:07:53.659064: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 AVX512F FMA
1125/1125 [-----] - 1s 799us/step - loss: 0.6438 - acc: 0.6329 - val_loss: 0.5123 - val_acc: 0.8347
Epoch 2/20
1125/1125 [-----] - 1s 585us/step - loss: 0.4935 - acc: 0.7920 - val_loss: 0.3942 - val_acc: 0.8880
Epoch 3/20
832/1125 [-----] - ETA: 0s - loss: 0.4514 - acc: 0.8017
```

Σχήμα A.3 – Εκπαίδευση και Επαλήθευση Μοντέλων Κατηγοριοποίησης Οχημάτων

Στο υποκεφάλαιο αυτό παρουσιάζονται τα αποτελέσματα των 6 ανιχνευτών οχημάτων. Σε κάθε ανιχνευτή οχημάτων δόθηκε ως είσοδος η εικόνα την οποία πρέπει να ανιχνεύσει για να βρει οχήματα και το μοντέλο αναγνώρισης και ανίχνευσης οχημάτων που παράχθηκε από τους κατηγοριοποιητές (6.2.1). Πιο συγκεκριμένα, τα αποτελέσματα αυτά λήφθηκαν μετά από εκπαίδευση, έλεγχο και επικύρωση των αντίστοιχων κατηγοριοποιητών με τους αντίστοιχους ανιχνευτές οχημάτων. Παραδείγματος χάρη, αν ο κατηγοριοποιητής οχημάτων, (παραγωγός μοντέλων αναγνώρισης και ανίχνευσης οχημάτων), χρησιμοποιεί το δίκτυο VGG16 στην φάση εκπαίδευσης, ελέγχου και επικύρωσης τότε και ο ανιχνευτής οχημάτων χρησιμοποιεί το ίδιο δίκτυο (VGG16) στην φάση εκπαίδευσης και ελέγχου. Οι ανιχνευτές οχημάτων έχουν εκπαιδευτεί και ελεγχθεί με τα δίκτυα ResNet50, VGG16, ShallowNet, MobileNet, VGG16 Transfer Learning και MobileNet Transfer Learning.

Παράρτημα Β – Τιμές που Δοκιμάστηκαν στους Ανιχνευτές Οχημάτων

Επιλεγμένα Πειράματα με Τιμές που Δοκιμάστηκαν στους Ανιχνευτές Οχημάτων:

Simple Detection Drone MobileNet Logistic Regression						
Variables:						
Experiment Number	Network	INPUT SIZE	PYR SCALE	WIN STEP	ROI SIZE	BATCH SIZE
1	VGG16	(350, 350)	1.5	16	(224, 224)	128
2	VGG16	(350, 350)	1.5	8	(224, 224)	128
4	VGG16	(350, 350)	1.25	8	(224, 224)	128
3	VGG16	(350, 350)	1.5	8	(224, 224)	64
7	VGG16	(350, 350)	1.22	8	(224, 224)	64
11	VGG16	(350, 350)	1.18	16	(224, 224)	256
15	VGG16	(350, 350)	1.03	16	(224, 224)	256
16	VGG16	(350, 350)	1.5	2	(224, 224)	256
19	VGG16	(350,350)	1.27	16	(224, 224)	256
25	VGG16	(350,350)	1.3	16	(224, 224)	256
26	VGG16	(350,350)	1.24	16	(224, 224)	256
1	ShallowNet	(350,350)	1.17	16	(32, 32)	256
2	ShallowNet	(350,350)	1.22	16	(32, 32)	256
3	ShallowNet	(350,350)	1.2	16	(32, 32)	256
15	ShallowNet	(350,350)	1.15	16	(32, 32)	256
17	ShallowNet	(350,350)	1.10	16	(32, 32)	256
19	ShallowNet	(350,350)	1.08	16	(32, 32)	256
21	ShallowNet	(350,350)	1.02	16	(32, 32)	256
23	ShallowNet	(350,350)	1.27	16	(32, 32)	256
24	ShallowNet	(350,350)	1.0	16	(32, 32)	512
40	ShallowNet	(350,350)	1.15	16	(32, 32)	64

41	ShallowNet	(350,350)	1.25	16	(32, 32)	64
44	ShallowNet	(350,350)	1.35	16	(32, 32)	64
53	ShallowNet	(350,350)	1.5	16	(32, 32)	64
59	ShallowNet	(350,350)	1.20	16	(32, 32)	64
6	MobileNet	(350,350)	1.23	8	(224, 224)	64
9	MobileNet	(350,350)	1.21	8	(160, 160)	64
14	MobileNet	(350,350)	1.21	4	(192, 192)	64
15	MobileNet	(350,350)	1.21	16	(192, 192)	64
16	MobileNet	(350,350)	1.21	8	(192, 192)	64
17	MobileNet	(350,350)	1.5	16	(192, 192)	64
18	MobileNet	(350,350)	1.23	16	(192, 192)	64
19	MobileNet	(350,350)	1.23	8	(192, 192)	64
20	MobileNet	(350,350)	1.23	4	(192, 192)	64
21	MobileNet	(350,350)	1.15	4	(192, 192)	64
26	MobileNet	(350,350)	1.05	4	(192, 192)	64
27	MobileNet	(350,350)	1.05	8	(192, 192)	64
28	MobileNet	(350,350)	1.35	8	(192, 192)	64
29	MobileNet	(350,350)	1.45	8	(192, 192)	64
33	MobileNet	(350,350)	1.05	4	(128, 128)	64
34	MobileNet	(350,350)	1.05	16	(128, 128)	64
36	MobileNet	(350,350)	1.15	16	(160, 160)	64
38	MobileNet	(350,350)	1.5	8	(160, 160)	64
39	MobileNet	(350,350)	1.25	8	(160, 160)	64
40	MobileNet	(350,350)	1.75	8	(160, 160)	64
41	MobileNet	(350,350)	1.75	4	(160, 160)	64
42	MobileNet	(350,350)	2.0	4	(160, 160)	64

Παράρτημα Γ – Κώδικας

Κώδικας #1

Όνομα Αρχείου: ftheadnet.py

Διεύθυνση Μονοπατιού: /home/pyimagesearch/nn/covn/ftheadnet.py

```
1 # import the necessary packages
2 from keras.layers.core import Dropout
3 from keras.layers.core import Flatten
4 from keras.layers.core import Dense
5
6 class FCHeadNet:
7     @staticmethod
8     def build(baseModel, classes, D):
9         # initialize the head model that will be placed on top of
10        # the base, then add a FC layer
11        headModel = baseModel.output
12        headModel = Flatten(name="flatten")(headModel)
13        headModel = Dense(D, activation="relu")(headModel)
14        headModel = Dropout(0.5)(headModel)
15
16        # add a softmax layer
17        headModel = Dense(classes, activation="softmax")(headModel)
18
19        # return the model
20        return headModel
```