

Ατομική Διπλωματική Εργασία

**ΣΥΣΤΗΜΑ ΑΥΤΟΜΑΤΗΣ ΑΝΑΓΝΩΡΙΣΗΣ ΠΙΝΑΚΙΔΩΝ
ΚΥΚΛΟΦΟΡΙΑΣ**

Ανδρέας Χατζηγεωργίου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2020

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Σύστημα Αυτόματης Αναγνώρισης Πινακίδων Κυκλοφορίας

Ανδρέας Χατζηγεωργίου

Επιβλέπων Καθηγητής
Δρ. Παττίχης Κωνσταντίνος

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

Μάιος 2020

Ευχαριστίες

Εκφράζω τις θερμές μου ευχαριστίες μου στον επιβλέπον καθηγητή μου κύριο Κωνσταντίνο Παττίχη για την καθοδήγηση, την υποστήριξη καθώς και την εμπιστοσύνη που μου επέδειξε κατά την εκπόνησης της παρούσας διπλωματικής εργασίας.

Επιπρόσθετα, θα ήθελα να ευχαριστήσω την οικογένεια μου για την στήριξη και την πίστη που επέδειξαν προς το πρόσωπο μου καθ' όλη τη διάρκεια των σπουδών μου, παρέχοντας μου παράλληλα στήριξη και παρότρυνση ώστε να πετύχω τους στόχους και τα όνειρα μου.

Επιπλέον, ευχαριστώ θερμά τους φίλους μου, οι οποίοι πάντοτε βρίσκονται στο πλάι μου στηρίζοντας το κάθε μου βήμα.

Περίληψη

Τα συστήματα αναγνώρισης πινακίδων κυκλοφορίας (LPR - License Plate Recognition), είναι μια από τις βασικότερες τεχνολογίες που αξιοποιούνται για την δημιουργία ενός έξυπνου κυκλοφοριακού συστήματος καθώς και σε πλήθος εφαρμογών εντός της καθημερινότητας μας.

Στόχος της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ενός πληροφοριακού συστήματος αναγνώρισης πινακίδων κυκλοφορίας, με σκοπό την ταυτοποίηση οχημάτων. Το σύστημα αυτό θα εντοπίζει τους αριθμούς εγγραφής οχημάτων, από εικόνα την οποία θα παρέχει ο χρήστης στο σύστημα και στην συνέχεια θα επιστρέφει στον χρήστη τους αριθμούς εγγραφής και τα στοιχεία του ιδιοκτήτη.

Για τον σχεδιασμό και την υλοποίηση του LPR συστήματος, έχει χρησιμοποιηθεί η γλώσσα προγραμματισμού Python. Για τον εντοπισμό και την αναγνώριση των πινακίδων κυκλοφορίας εντός των εικόνων, αξιοποιήθηκε ο αλγόριθμος YOLOv3, ενώ για την οπτική αναγνώριση των χαρακτήρων που αποτελούν την πινακίδα κυκλοφορίας, αξιοποιήθηκε το OCR Tesseract.

Για την αξιολόγηση της λειτουργικότητας και αποτελεσματικότητας του συστήματος, χρησιμοποιήθηκαν εικόνες οχημάτων από διάφορες λήψεις, σε όλες όμως τις περιπτώσεις χρησιμοποιήθηκαν πινακίδες κυκλοφορίας της Κυπριακής Δημοκρατίας. Διαφάνηκε ότι το σύστημα εντοπίζει και αναγνωρίζει έναν μεγάλο αριθμό πινακίδων κυκλοφορίας, όμως καθοριστικό ρόλο για την αποτελεσματικότητα του συστήματος, έχει η ποιότητα της εικόνας καθώς και η κλίση της πινακίδας εντός της εικόνας. Βάσει των αποτελεσμάτων, εξάχθηκαν συμπεράσματα καθώς επίσης και εισηγήσεις για μελλοντική ανάπτυξη του συστήματος.

Λέξεις κλειδιά:

Αυτόματο Σύστημα αναγνώρισης πινακίδων κυκλοφορίας (ANPR), LPR, Συνελικτικά Νευρωνικά Δίκτυα, YOLOv3, Tesseract OCR

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Εισαγωγικά Στοιχεία	1
	1.2 Σκοπός & Αναγκαιότητα της παρούσας διπλωματικής εργασίας	2
	1.3 Δομή Διπλωματικής Εργασίας	3
Κεφάλαιο 2	Θεωρητικό Υπόβαθρο	5
	2.1 Εισαγωγή	5
	2.2 Μηχανική Μάθηση	6
	2.3 Βαθιά Μηχανική Μάθηση	8
	2.4 Τεχνητά Νευρωνικά Δίκτυα	10
	2.5 Συνελικτικά Νευρωνικά Δίκτυα	13
Κεφάλαιο 3	Συστήματα Αναγνώρισης Πινακίδων Κυκλοφορίας.....	18
	3.1 Ανάπτυξη Συστημάτων Αναγνώρισης Πινακίδων Κυκλοφορίας	18
	3.2 Χαρακτηριστικά Πινακίδων Κυκλοφορίας για την περίπτωση της Κυπριακής Δημοκρατίας	22
Κεφάλαιο 4	Σχεδιασμός και Ανάπτυξη συστήματος αυτόματης αναγνώρισης πινακίδων κυκλοφορίας.....	26
	4.1 Εξακρίβωση Αναγκών και Καθορισμός Απαιτήσεων	26
	4.2 Καθορισμός Προδιαγραφών	27
	4.3 Εργαλεία & Τεχνικές που θα χρησιμοποιηθούν	28
	4.4 Σχεδιασμός και Υλοποίηση Συστήματος	31
	4.5 Έλεγχος Συστήματος	42
	4.6 Αποτελέσματα	44
	4.7 Συσχετισμός με άλλες μελέτες	45
	4.8 Δυσκολίες	47

ΚΕΦΑΛΑΙΟ 5	Συμπεράσματα και Μελλοντική εργασία.....	48
5.1	Συμπεράσματα	48
5.2	Μελλοντική εργασία	49
Βιβλιογραφία		51

Κεφάλαιο 1

Εισαγωγή

1.1 Εισαγωγικά Στοιχεία	1
1.2 Σκοπός & Αναγκαιότητα της παρούσας διπλωματικής εργασίας	2
1.3 Δομή διπλωματικής εργασίας	3

1.1 Εισαγωγικά Στοιχεία

Ο γρήγορος ρυθμός ο οποίος χαρακτηρίζει πλέον την κοινωνία μας στη σημερινή εποχή, έχει οδηγήσει την πλειοψηφία του πληθυσμού στην απόκτηση και χρήση ως μέσο μεταφοράς, το προσωπικό του αυτοκίνητο. Δεδομένου ότι κάθε όχημα έχει πινακίδες με έναν μοναδικό αριθμό εγγραφής, κρίνεται απαραίτητη η ανάγκη ανάπτυξης συστημάτων αυτόματης αναγνώρισης πινακίδων κυκλοφορίας, με σκοπό τη δημιουργία ενός έξυπνου κυκλοφοριακού συστήματος.

Η τεχνολογία των συστημάτων αναγνώρισης πινακίδων κυκλοφορίας ξεκίνησε το 1976 στο τμήμα Τεχνολογικής Ανάπτυξης της Αστυνομίας του Ηνωμένου Βασιλείου και ξεκίνησε να αξιοποιείται το 1979. Το 1981 με την αξιοποίηση της τεχνολογίας αυτής, καταγράφηκε η πρώτη σύλληψη για κλοπή αυτοκινήτου, ενώ το 2005 η χρήση της τεχνολογίας αυτής συνέβαλε στην εξιχνίαση ανθρωποκτονίας (Maulin & Bhalodia, 2014: 99). Ακολούθως ξεκίνησε και η εμπορική διάθεση της σε εταιρίες, επεκτείνοντας την λειτουργία του σε πλήθος εφαρμογών πέραν του αστυνομικού ελέγχου, όπως για παράδειγμα, τον έλεγχο διέλευσης διοδίων, στις μη-ελεγχόμενες θέσεις στάθμευσης, στον έλεγχο ασφάλειας απαγορευμένων περιοχών, στην επιβολή της νομοθεσίας για τον περί οδικής κυκλοφορίας νόμο, στην αποσυμφόρηση του κυκλοφοριακού δικτύου και στην αυτόματη είσπραξη διοδίων (Chang et al., 2004 : 42; Draghici, 1997: 2-4). Φυσικά οι μέθοδοι ανάπτυξης τέτοιων συστημάτων ποικίλουν και ανάλογα της λειτουργίας του κάθε συστήματος διαφέρουν μεταξύ τους.

1.2 Σκοπός & Αναγκαιότητα της παρούσας διπλωματικής εργασίας

Στόχος της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ενός πληροφοριακού συστήματος αναγνώρισης πινακίδων κυκλοφορίας, με σκοπό την ταυτοποίηση οχημάτων μέσω της πινακίδας κυκλοφορίας. Το σύστημα αυτό θα εντοπίζει τους αριθμούς εγγραφής οχημάτων, από εικόνα την οποία θα παρέχει στο σύστημα ο χρήστης και στην συνέχεια θα επιστρέφει στον χρήστη τα στοιχεία του οχήματος καθώς και στοιχεία του κατόχου.

Η επιλογή του συγκεκριμένου θέματος πρόεκυψε λόγω του ότι, έχω εξ ιδίας αντίληψη για την αναγκαιότητα και την χρησιμότητα ενός τέτοιου πληροφοριακού συστήματος στα Σώματα Ασφαλείας της Δημοκρατίας, με απώτερο σκοπό την ενίσχυση του ελέγχου κυκλοφορίας καθώς και για την καταπολέμηση του εγκλήματος. Επιπρόσθετα, ακόμη ένας λόγος, είναι ότι τα συστήματα αυτά έχουν ενσωματωθεί σε διάφορους τομείς της καθημερινότητας μας, γεγονός που οδήγησε στην αύξηση ενδιαφέροντος και στον τομέα της έρευνας και της ανάπτυξης της βιβλιογραφίας, χωρίς όμως ακόμη να εκμηδενιστούν οι περιορισμοί όπου επηρεάζουν την αποτελεσματικότητα των συστημάτων αυτών (Rio-Alvarez, et al, 2019:1).

Τα πλείστα LPR συστήματα τα οποία έχουν αναπτυχθεί και στο παρελθόν, λειτουργούν σε ιδιαίτερες συνθήκες καθώς και σε συγκεκριμένες και περιορισμένες περιπτώσεις. Οι παράγοντες που επηρεάζουν την λειτουργία του συστήματος είναι αρκετοί, κάποιιοι από αυτούς είναι ο φωτισμός της εικόνας, η ταχύτητα του οχήματος, η πορεία όπου ακολουθούσε το όχημα, το φόντο της φωτογραφίας (Chang et al., 2004: 42), οι καιρικές συνθήκες ή ακόμη και η εσκεμμένη αλλοίωση των πινακίδων κυκλοφορίας. Ένα σύγχρονο σύστημα είναι απαραίτητο να μην χαρακτηρίζεται από τέτοιου είδους περιορισμούς, εφόσον η σημερινή τεχνολογία μπορεί να μας παρέχει τα απαραίτητα εργαλεία και εφόδια για εξασφάλιση καλύτερου αποτελέσματος.

Η αναγνώριση αριθμών εγγραφής οχημάτων λοιπόν, αποτελεί αναπόσπαστο κομμάτι διάφορων εφαρμογών σε διάφορους τομείς και πτυχές της κοινωνίας. Αυτός άλλωστε είναι και ο λόγος, που πολλές εταιρίες επενδύουν στην ανάπτυξη τέτοιων συστημάτων. Η ραγδαία εξέλιξη της τεχνολογίας προϋποθέτει ότι τέτοια συστήματα πρέπει να

αναβαθμίζονται και να εκσυγχρονίζονται συστηματικά, ώστε να εκμηδενίσουν περιορισμούς αλλά και τυχόν προβλήματα που παρουσιάζουν τα συστήματα.

Κατ' επέκταση, σκοπός των σύγχρονων LPR συστημάτων είναι η γρήγορη επεξεργασία και η αποτελεσματικότητα υπό διάφορες συνθήκες, οι οποίες μπορεί μέχρι και στο παρόν στάδιο, να θέτουν περιορισμούς στην λειτουργικότητα των συστημάτων (Maulin & Bhalodia, 2014: 99). Οπότε, κύριος σκοπός της παρούσας διπλωματικής εργασίας δεν είναι μόνο η ανάπτυξη ενός τέτοιου συστήματος, αλλά ενός συστήματος το οποίο θα ανταποκρίνεται στις δυνατότητες της σύγχρονης τεχνολογίας και τις τοπικές απαιτήσεις.

1.3 Δομή διπλωματικής εργασίας

Η ατομική διπλωματική εργασία είναι οργανωμένη στα πιο κάτω κεφάλαια:

Στο κεφάλαιο 1, γίνεται μια εισαγωγή στο θέμα με το οποίο θα ασχοληθεί η παρούσα διπλωματική εργασία, καθώς επίσης, γίνεται αναφορά στον σκοπό και την αναγκαιότητα της εργασίας αυτής.

Στο κεφάλαιο 2, παρουσιάζεται το θεωρητικό υπόβαθρο στο οποίο θα στηριχθούμε για την ανάπτυξη του συστήματος. Πιο συγκεκριμένα γίνεται αναφορά στην Μηχανική και την Βαθιά Μάθηση καθώς και στα Τεχνητά Νευρωνικά Δίκτυα και τα Συνελικτικά Νευρωνικά Δίκτυα.

Στο κεφάλαιο 3, γίνεται αναφορά στην ανάπτυξη καθώς και στον τρόπο που λειτουργούν τα συστήματα αυτόματης αναγνώρισης πινακίδων κυκλοφορίας, παρουσιάζοντας την διαδικασία που ακολουθεί ένα τέτοιο σύστημα καθώς και τις μεθόδους και τεχνικές που αξιοποιούνται σε αυτό. Επιπλέον, για τον λόγο ότι για την ανάπτυξη ενός λειτουργικού συστήματος είναι απαραίτητη η δόμηση του με βάση τα χαρακτηριστικά των πινακίδων όπου θα αναγνωρίζει, γίνεται αναφορά στα χαρακτηριστικά των πινακίδων κυκλοφορίας της Κυπριακής Δημοκρατίας.

Στο κεφάλαιο 4, περιγράφεται η όλη διαδικασία που ακολουθήθηκε για τον σχεδιασμό και την ανάπτυξη του δικού μας συστήματος αυτόματης αναγνώρισης πινακίδων κυκλοφορίας. Πιο συγκεκριμένα, γίνεται αναφορά στην εξακρίβωση αναγκών και στον καθορισμό απαιτήσεων, στον καθορισμό των προδιαγραφών του συστήματος, στα εργαλεία και τις τεχνικές που θα χρησιμοποιηθούν, στον σχεδιασμό και τον τρόπο υλοποίησης, στον έλεγχο του συστήματος, τις δυσκολίες που αντιμετωπίσαμε καθώς και τα αποτελέσματα του συστήματος.

Στο κεφάλαιο 5, γίνεται αναφορά στα συμπεράσματα που εξήχθησαν καθώς και σε εισηγήσεις μελλοντικής εργασίας και αναβάθμισης του συστήματος.

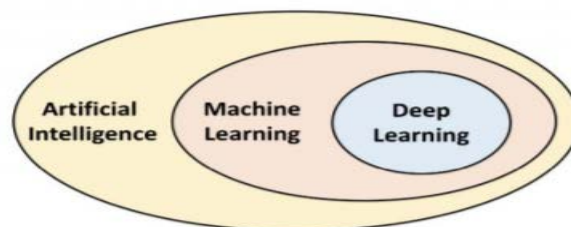
Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

2.1 Εισαγωγή	5
2.2 Μηχανική Μάθηση	6
2.3 Βαθιά Μηχανική Μάθηση	8
2.4 Τεχνητά Νευρωνικά Δίκτυα	10
2.5 Συνελκτικά Νευρωνικά Δίκτυα	13

2.1 Εισαγωγή

Η τεχνητή νοημοσύνη (Artificial intelligence) επιδιώκει να παρέχει στον υπολογιστή την ανθρώπινη ευφυΐα, να είναι δηλαδή έξυπνο και να μπορεί να λαμβάνει αποφάσεις, η μηχανική και η βαθιά μάθηση, είναι οι τεχνολογίες οι οποίες βοηθούν την τεχνική νοημοσύνη να το επιτύχουν. Η μηχανική μάθηση (Machine learning) είναι αυτή που εκπαιδεύει τις μηχανές για το πώς να μαθαίνουν. Η βαθιά μάθηση (Deep Learning), είναι μια βελτιωμένη εκδοχή της μηχανικής μάθησης, η οποία βοηθάει ως προς την αύξηση του «επιπέδου» του μαθησιακού περιβάλλοντος. Και οι δύο μορφές μάθησης έχουν σημαντικό ρόλο στην αναβάθμιση των συστημάτων ώστε να μπορούν να λαμβάνουν αποφάσεις και να κάνουν προβλέψεις χωρίς ανθρώπινη παρέμβαση (Chahal & Gulia, 2019: 4910).



Σχ.1: Σχέση ανάμεσα στην Τεχνητη Νοημοσύνη, την Μηχανική Μάθηση και την Βαθιά Μάθηση

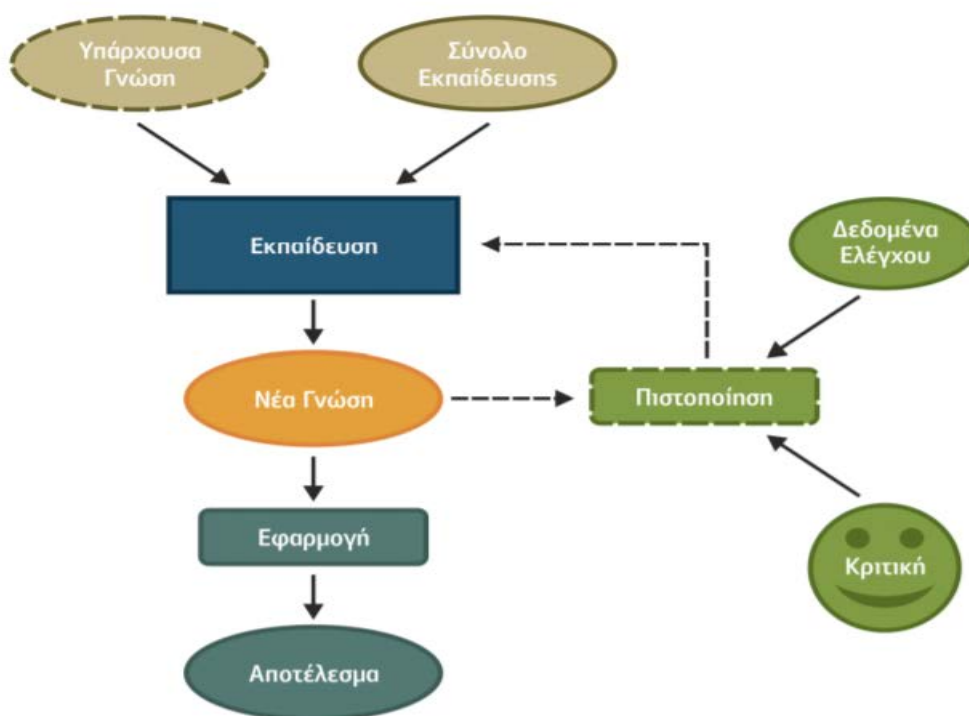
(Chahal & Gulia, 2019: 4910)

2.2 Μηχανική Μάθηση

Η μηχανική μάθηση είναι μια περιοχή του ευρύτερου πεδίου της τεχνητής νοημοσύνης (Artificial Intelligence), η οποία ασχολείται με αλγορίθμους που παρέχουν στα συστήματα την ικανότητα να μαθαίνουν μέσω παραδειγμάτων, δεδομένων και της εμπειρίας τους. Σκοπός δηλαδή, της μηχανικής μάθησης είναι η δημιουργία μηχανών ικανών να μαθαίνουν και να βελτιώνουν την απόδοσή λειτουργίας τους, μέσω αξιοποίησης προηγούμενης γνώσης και εμπειρίας, χωρίς να ακολουθούν έναν ρητό προγραμματισμό (Γεωργούλη, 2015: 127; Chahal & Gulia, 2019: 4910).

Πιο συγκεκριμένα, είναι η ικανότητα ενός υπολογιστικού συστήματος να δημιουργεί μοντέλα ή πρότυπα από ένα σύνολο δεδομένων, γεγονός που φέρει ως αποτέλεσμα την βελτίωση της απόδοσης του. Για τη δημιουργία ενός προτύπου, οι αλγόριθμοι μηχανικής μάθησης, αναπτύσσουν στατιστικά μοντέλα, βασιζόμενα σε ένα σύνολο δεδομένων, προϋπάρχουσας γνώσης και νέων δεδομένων, προκειμένου τα δεδομένα να ταξινομηθούν σε κατηγορίες ή να αναγνωρίσουν τυχόν τυποποιημένες μορφές σε αυτά, με απώτερο σκοπό τη λήψη αποφάσεων από το σύστημα, με την ελάχιστη ανθρώπινη παρέμβαση (Chahal & Gulia, 2019: 4910; Γεωργούλη, 2015: 127).

Σημαντικό ρόλο για τους αλγόριθμους της μηχανικής μάθησης, έχει η εκπαίδευση τους, κατά την οποία ο αλγόριθμος χρησιμοποιεί ως είσοδο ένα σύνολο δεδομένων εκπαίδευσης (training set) που του παρέχει ο χρήστης, με σκοπό τη δημιουργία της αρχικής γνώσης. Σε μετέπειτα στάδιο, για την ενίσχυση της γνώσης του, ο αλγόριθμος συνδυάζοντας την υπάρχουσα γνώση με κάποια νέα δεδομένα εισόδου, δημιουργεί μια καινούρια γνώση. Κατά το στάδιο αυτό, ο αλγόριθμος έχει την δυνατότητα να χρησιμοποιήσει λιγότερο ή περισσότερο την προϋπάρχουσα γνώση του, είτε να μην τη χρησιμοποιήσει καθόλου. Την εκπαίδευση ακολουθεί η φάση της πιστοποίησης της παραγόμενης νέας γνώσης. Συνήθως, η πιστοποίηση πραγματοποιείται αρχικά από τον ίδιο τον αλγόριθμο μέσω διαδικασιών ανάκλησης (recall) με τη βοήθεια δεδομένων ελέγχου (test data) και στη συνέχεια, μέσω κριτικής που κάνει ο χρήστης, βάσει των γνώσεων που διαθέτει για το πρόβλημα που επιχειρεί να λύσει ο αλγόριθμος. Τέλος, η νέα γνώση δίνεται προς χρήση σε εφαρμογές στις οποίες είναι απαραίτητη, για να λυθούν πραγματικά προβλήματα (Γεωργούλη, 2015: 129).



Σχ.2: Φάσεις Μηχανικής Μάθησης

(Γεωργούλη, 2015: 129)

Ο τομέας της Μηχανικής Μάθησης έχει αναπτύξει τρεις τρόπους μάθησης, ανάλογους με τους τρόπους με τους οποίους μαθαίνει ο άνθρωπος, την επιβλεπόμενη μάθηση, την μη επιβλεπόμενη μάθηση και την ενισχυτική μάθηση. Η επιβλεπόμενη μάθηση (Supervised Learning) είναι η διαδικασία όπου ο αλγόριθμος κατασκευάζει μια συνάρτηση που απεικονίζει δεδομένες εισόδους (σύνολο εκπαίδευσης) με γνωστές επιθυμητές εξόδους, με στόχο τη γενίκευση της συνάρτησης αυτής, ώστε να λειτουργεί και για εισόδους με άγνωστη έξοδο. Η επιβλεπόμενη μάθηση χρησιμοποιείται κυρίως σε προβλήματα ταξινόμησης (Classification), πρόγνωσης (Prediction) και Διερμηνείας (Interpretation). Η μη επιβλεπόμενη μάθηση (Unsupervised Learning) είναι η διαδικασία όπου ο αλγόριθμος καλείται να κατασκευάσει ένα μοντέλο για κάποιο σύνολο εισόδων χωρίς να γνωρίζει τις επιθυμητές εξόδους. Η μη επιβλεπόμενη μάθηση χρησιμοποιείται κυρίως σε προβλήματα ανάλυσης συσχετισμών (Association Analysis) και ομαδοποίησης (Clustering). Η ενισχυτική μάθηση (Reinforcement Learning) είναι η διαδικασία όπου ο αλγόριθμος μαθαίνει μια στρατηγική ενεργειών μέσα από άμεση αλληλεπίδραση με το περιβάλλον. Η ενισχυτική μάθηση, χρησιμοποιείται κυρίως σε προβλήματα σχεδιασμού (Planning) (Γεωργούλη, 2015: 128).

2.3 Βαθιά Μηχανική Μάθηση

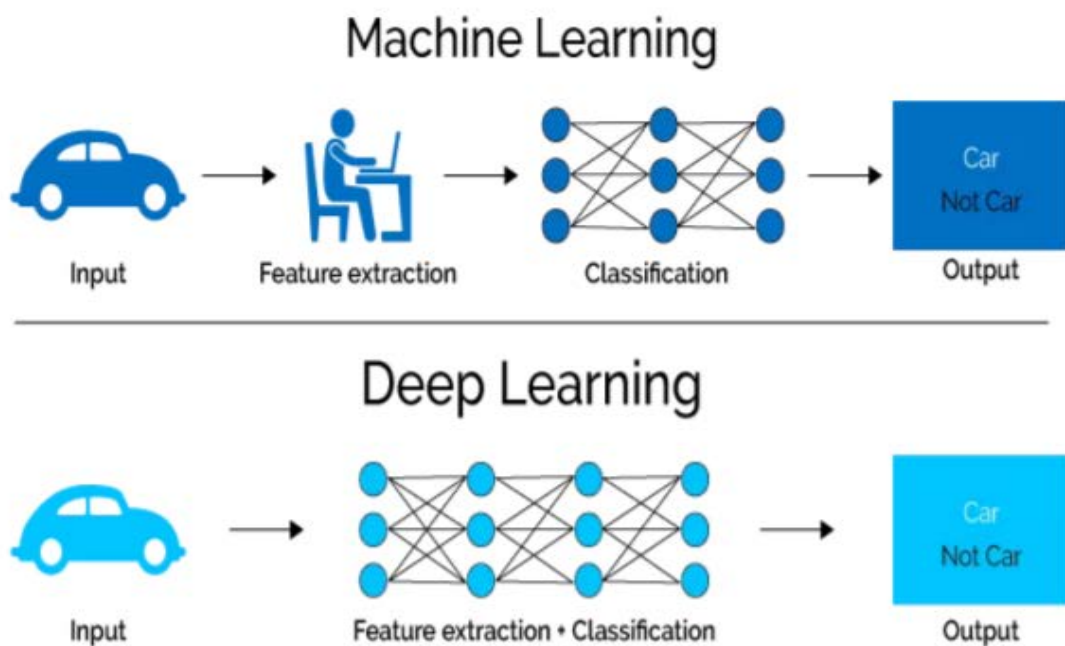
Οι μέθοδοι μηχανικής μάθησης κατηγοριοποιούνται σε δύο κατηγορίες, την ρηχή μάθηση (Shallow learning) και την βαθιά μάθηση (Deep Learning). Η ρηχή μάθηση, χρησιμοποιεί για την επεξεργασία των δεδομένων, νευρωνικό δίκτυο ενός επιπέδου ενώ η βαθιά μάθηση, νευρωνικά δίκτυα πολλαπλών επιπέδων, πράγμα που οδηγεί στην επεξεργασία των δεδομένων εις βάθος (Chahal & Gulia, 2019: 4910). Ουσιαστικά, η βαθιά μηχανική μάθηση είναι η διαδικασία εφαρμογής «μαθησιακών διεργασιών» σε νευρωνικά δίκτυα πολλών επιπέδων.

Κατ' επέκταση, η βαθιά μηχανική μάθηση, χρησιμοποιεί πολλαπλά στρώματα επεξεργασίας των δεδομένων με αποτέλεσμα να διεκπεραιώνει καλύτερα και αποδοτικότερα την επίλυση σύνθετων προβλημάτων. Για τον λόγο αυτόν, είναι πλέον ευρέως διαδεδομένη καθώς έχει βελτιώσει σε σημαντικό βαθμό την απόδοση πολλών συστημάτων όπως είναι για παράδειγμα μηχανική όραση, η αναγνώριση και σύνθεση φωνής, η ταξινόμηση, η οπτική αναγνώριση αντικειμένων και πολλών άλλων. Χαρακτηρίζεται ως πολλά υποσχόμενη ως προς τον δρόμο της έρευνας για την εξαγωγή σύνθετων αναπαραστάσεων δεδομένων (Chahal & Gulia, 2019: 4912; Najafabadi, 2015:2)

Ο τρόπος με τον οποίο λειτουργεί η βαθιά μηχανική μάθηση, βασίζεται στον τρόπο με τον οποίο λειτουργεί ο ανθρώπινος εγκέφαλος και οι νευρώνες για την επεξεργασία πληροφοριών και σημάτων. Η αρχιτεκτονική της, χαρακτηρίζεται από πολλαπλά επίπεδα μη γραμμικής επεξεργασίας και αποτελείται από ένα επίπεδο εισόδου, ένα επίπεδο εξόδου και πολλά κρυφά επίπεδα. Το κάθε επίπεδο έχει ως στόχο την εξαγωγή ιεραρχικών χαρακτηριστικών από τα δεδομένα, το αποτέλεσμα της προηγούμενης διεργασίας είναι η είσοδος της επόμενης, με αποτέλεσμα να γίνεται μια ιεραρχική αναπαράσταση των δεδομένων και κατά συνέπεια, το σύστημα να επιτυγχάνει να μαθαίνει και να αναπαριστά τις πολύπλοκες έννοιες μέσα από την σύνθεση των ενδόμυχων χαρακτηριστικών που λήφθηκαν από το κάθε επίπεδο.

Επιπρόσθετα, σε αντίθεση με τις συμβατικές μεθόδους μηχανικής μάθησης, η βαθιά μηχανική μάθηση δεν χρειάζεται να «διδασχθεί» από κάποιο σύνολο δεδομένων

εκπαίδευσης που θα του παράσχει ο χρήστης, μπορεί να επεξεργαστεί απευθείας δεδομένα τα οποία θα του εισαχθούν. Επιπρόσθετα, σε σύγκριση με τις συμβατές μηχανικές μεθόδους, οι αλγόριθμοι βαθιάς μάθησης μπορούν να λειτουργούν με μεγάλη ακρίβεια, ταχύτητα και αποτελεσματικότητα σε μεγάλο όγκο δεδομένων. Ο μεγάλος αυτός όγκος δεδομένων φέρει ως αποτέλεσμα, την συνεχή εκπαίδευση του αλγόριθμου με νέα δεδομένα και ως εκ τούτου, μέσω της εμπειρίας του επιτυγχάνει την συνεχή βελτίωση έτσι ώστε να παράγει καλύτερα αποτελέσματα (Chahal & Gulia, 2019: 4913 - 4914).



Σχ.3: Διαδικασία που ακολουθεί η Μηχανική Μάθηση και η Βαθιά Μάθηση

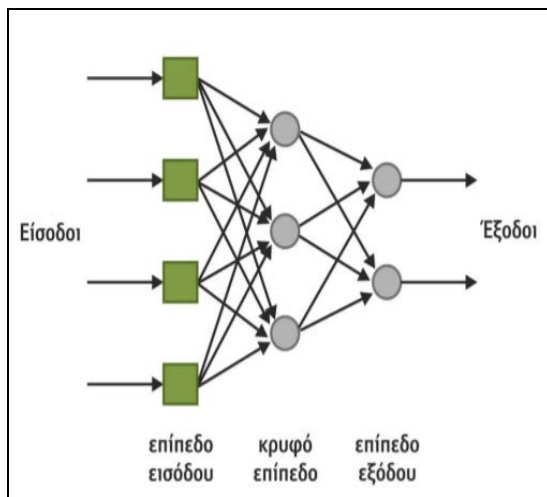
(Chahal & Gulia, 2019: 4913)

2.3 Τεχνητά Νευρωνικά Δίκτυα

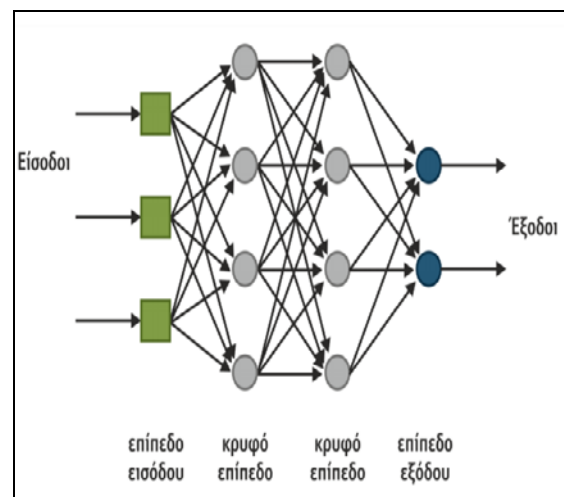
Τα τεχνητά νευρωνικά δίκτυα (Artificial Neural Networks) είναι μια σειρά αλγορίθμων μηχανικής μάθησης και είναι εμπνευσμένα από τον τρόπο με τον οποίο λειτουργούν τα βιολογικά νευρωνικά δίκτυα. Ξεκίνησαν ως μία προσπάθεια μοντελοποίησης της συμπεριφοράς του ανθρώπινου εγκεφάλου, ωστόσο σήμερα τα νευρωνικά δίκτυα χρησιμοποιούνται για να λύσουν κάθε είδους προβλήματα με ηλεκτρονικό υπολογιστή, η λειτουργία τους επιδιώκει τον συνδυασμό του τρόπου σκέψης του ανθρώπινου εγκεφάλου με τον αφηρημένο μαθηματικό τρόπο σκέψης (Mijwil & Esen & Alsaadi, 2019: 1).

Αποτελούνται από ένα αφηρημένο μοντέλο διασυνδεδεμένων νευρώνων, των οποίων η ειδική διάταξη και η σύνδεση τους μπορεί να αξιοποιηθεί στην μηχανική μάθηση και την τεχνητή νοημοσύνη. Επιπλέον μπορούν να αξιοποιηθούν για την επίλυση διαφόρων ειδών προβλημάτων και εφαρμογών σε διάφορους τομείς, κάποια από τα παραδείγματα για τα οποία γίνεται αξιοποίηση νευρωνικών δικτύων είναι η αναγνώριση εικόνας / φωνής / μοτίβου, η σύνθεση ομιλίας, προβλέψεις και προσομοιώσεις σύνθετων συστημάτων, συστήματα έγκαιρης προειδοποίησης, οικονομικά μοντέλα και πολλά άλλα, προηγουμένως όμως θα πρέπει να τύχουν κάποιας εκπαίδευσης έτσι ώστε να μπορέσουν να επιλύσουν το κάθε πρόβλημα (Mijwil & Esen & Alsaadi, 2019: 1).

Τα τεχνητά νευρωνικά δίκτυα αποτελούνται από μια συλλογή μεγάλου αριθμού συνδεδεμένων στοιχείων επεξεργασίας που ονομάζονται νευρώνες (Processing Units) οι οποίοι συνδέονται μεταξύ τους, λειτουργούν αρμονικά και είναι οργανωμένοι σε στρώματα / επίπεδα (Layers) (Πλέρου, 2012:131). Μπορεί να γίνει διάκριση μεταξύ νευρώνων εισόδου, κρυφών νευρώνων και νευρώνων εξόδου. Στην περίπτωση όπου υπάρχουν πολλά κρυμμένα επίπεδα, τα τεχνητά νευρωνικά δίκτυα χαρακτηρίζονται ως πολυεπίπεδα (Γεωργούλη,2015:161).



Σχ.4: Απλό τεχνητό νευρωνικό δίκτυο



Σχ.5: Πολυεπίπεδο Τεχνητό Νευρωνικό Δίκτυο

(Γεωργούλη, 2015: 160 - 161)

Οι νευρώνες εισόδου λαμβάνουν πληροφορίες με τη μορφή μοτίβων ή σημάτων από τον «έξω κόσμο», οι κρυμμένοι νευρώνες βρίσκονται μεταξύ των νευρώνων εισόδου και εξόδου και χαρτογραφούν εσωτερικά μοτίβα πληροφοριών, ενώ οι νευρώνες εξόδου μεταδίδουν πληροφορίες και σήματα στον «έξω κόσμο», ως αποτέλεσμα. Οι νευρώνες συνδέονται μεταξύ τους μέσω των αποκαλούμενων «άκρων», ενώ η έξοδος ενός νευρώνα, μπορεί να είναι η είσοδος του επόμενου, με αποτέλεσμα να δημιουργείται ένα δίκτυο νευρώνων (Mijwil & Esen & Alsaadi, 2019: 1). Κάθε τεχνητός νευρώνας αποτελείται από πολλές εισόδους και μία μόνο έξοδο, κάθε είσοδος “ζυγίζεται” με κάποιο βάρος (Γεωργούλη, 2015: 157). Οι συνδέσεις μεταξύ των νευρώνων διαφέρουν μεταξύ τους και η σπουδαιότητα τους προσδιορίζεται από τον συντελεστή βάρους (Πλέρου, 2012:131).

Για να χρησιμοποιηθεί ένα τεχνικό νευρωνικό δίκτυο, θα πρέπει πρώτα να περάσει από την διαδικασία εκπαίδευσης, η οποία ξεκινάει από μία κατάσταση στην οποία δεν υπάρχει καμία γνώση. Μέσω της εκπαίδευσης, αποκτάται η απαραίτητη «εμπειρία», η οποία αποθηκεύεται στα συνοπτικά βάρη, έτσι ώστε να επιτυγχάνεται η επίλυση των προβλημάτων. Κατά τη διάρκεια την εκπαίδευσης του νευρωνικού δικτύου, οι σταθμίσεις των συνδέσεων αλλάζουν, ανάλογα με τους εφαρμοσμένους κανόνες

μάθησης και τα ληφθέντα αποτελέσματα. Με βάση το δεδομένο μαθησιακό υλικό και τους κανόνες μάθησης, το νευρικό δίκτυο ζυγίζει τις συνδέσεις των νευρώνων μέχρι να αναπτύξει συγκεκριμένη «νοημοσύνη» (Mijwil & Esen & Alsaadi, 2019: 1; Πλέρου, 2012 :131).

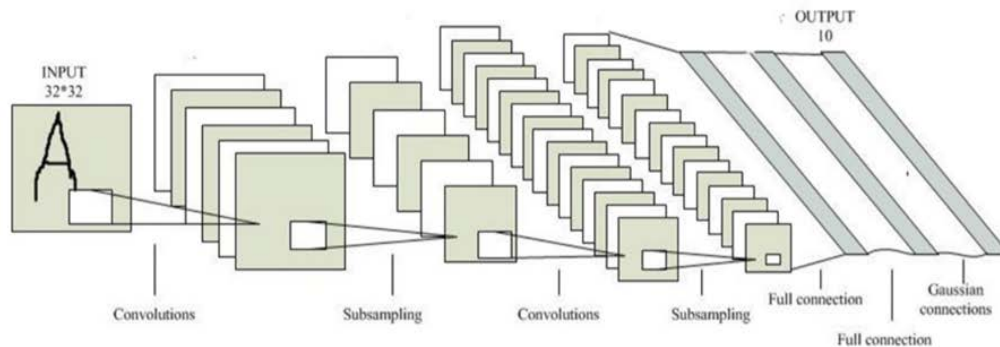
Συνεπώς, ως Τεχνικό νευρωνικό δίκτυο, ορίζεται ως ένας παράλληλος κατανεμημένος επεξεργαστής με κλίση στην αποθήκευση και την απόδοση εμπειρικής γνώσης. Η γνώση των τεχνικών νευρικών δικτύων, λαμβάνεται μέσω μιας διαδικασίας εκπαίδευσης, ενώ η αποθήκευση της γνώσης αυτής γίνεται μέσω των βαρών που υπάρχουν στις συνδέσεις μεταξύ των νευρώνων (Πλέρου, 2012: 132)

Η εκπαίδευση έχει ως βασικό στόχο να βρεθεί ένας τρόπος αλλαγής των συνδεσμικών βαρών που θα έχει ως αποτέλεσμα την αλλαγή της γενικής συμπεριφοράς του δικτύου με την αύξηση της ικανότητας του δικτύου να παρέχει στο μέλλον μία επιθυμητή έξοδο μετά από μία δεδομένη είσοδο. Η εκπαίδευση αυτή, μπορεί να διακριθεί στις περιπτώσεις, της εποπτευόμενης μάθησης και της μη εποπτευόμενης μάθησης. Στην εποπτευόμενη μάθηση, το δίκτυο τροφοδοτείται με ένα συγκεκριμένο επιθυμητό αποτέλεσμα και διάφορες πιθανές επιλογές εισόδου, με βάση τη συνεχή σύγκριση, το δίκτυο μαθαίνει να συνδέει κατάλληλα τους νευρώνες, ώστε να εντοπίζει το σωστό και επιθυμητό αποτέλεσμα και σε άλλα άγνωστα πρότυπα. Στην μη εποπτευόμενη μάθηση, το δίκτυο τροφοδοτείται με μια σειρά από πρότυπα χωρίς όμως να έχουν ετικέτες (δεν γνωρίζει τον στόχο) και καλείται να εξάγει συμπεράσματα από μόνο του. Συμπερασματικά, όταν η επιθυμητή έξοδος είναι εκ των προτέρων γνωστή λέμε ότι το δίκτυο μαθαίνει με επίβλεψη, αλλιώς μαθαίνει χωρίς επίβλεψη (Mijwil & Esen & Alsaadi, 2019: 1; Γεωργούλη, 2015:166).

2.4 Συνελικτικά Νευρωνικά Δίκτυα

Τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks - CNNs), είναι μια κατηγορία Τεχνητών Νευρωνικών Δικτύων, συνεπώς αποτελούνται από νευρώνες οι οποίοι επιδιώκουν την βελτίωση τους μέσω της μάθησης και της εμπειρίας τους (O'Shea & Nash, 2015:2). Τα Συνελικτικά Νευρωνικά Δίκτυα, είναι μια προσέγγιση βαθιάς μάθησης η οποία χρησιμοποιείται ευρέως για την επίλυση σύνθετων προβλημάτων (Indolia et al., 2018:679). Έχει σημειωθεί ότι έχουν εφαρμοστεί με επιτυχία σε εφαρμογές υπολογιστικής όρασης (Kamilaris & Prenafeta-Boldú, 2018:1) και χρησιμοποιούνται κυρίως για την επίλυση δύσκολων εργασιών αναγνώρισης προτύπων βάσει εικόνας και βίντεο (O'Shea & Nash, 2015:1). Για τον λόγο ότι τα δίκτυα αυτά αξιοποιούνται κυρίως για δεδομένα εικόνας και βίντεο, η αρχιτεκτονική των δικτύων έχει διαμορφωθεί με τρόπο ο οποίος είναι κατάλληλος για την δομή τέτοιου είδους δεδομένων (O'Shea & Nash, 2015:3; Arel & Rose & Karnowski, 2010:14).

Τα Συνελικτικά Νευρωνικά Δίκτυα, αποτελούνται από επίπεδο εισόδου και εξόδου καθώς και από πολλαπλά κρυμμένα επίπεδα, τα οποία χωρίζονται σε τρεις τύπους, τα Συνελικτικά Επίπεδα (convolutional layers), τα Επίπεδα Συγκέντρωσης (pooling / subsampling layers) και τα Πλήρως Συνδεδεμένα Επίπεδα (fully-connected layers) (Sakib et al., 2018:1; Zhang, 2018: 435). Τα διάφορα αυτά επίπεδα, αποτελούνται από νευρώνες οργανωμένους σε τρεις διαστάσεις, το ύψος, το πλάτος και το βάθος, γνωστά ως χωρική διάσταση της εισόδου (Sakib et al., 2018:2). Τα συνελικτικά επίπεδα έχουν σκοπό την εξαγωγή χαρακτηριστικών από τις εικόνες εισόδου των οποίων οι διαστάσεις μειώνονται από τα επίπεδα συγκέντρωσης, ενώ τα πλήρως συνδεδεμένα στρώματα λειτουργούν ως ταξινομητές (Kamilaris & Prenafeta-Boldú, 2018:3).



Σχ.6: Αρχιτεκτονική δομή ενός Συνελεκτικού Νευρωνικού Δικτύου

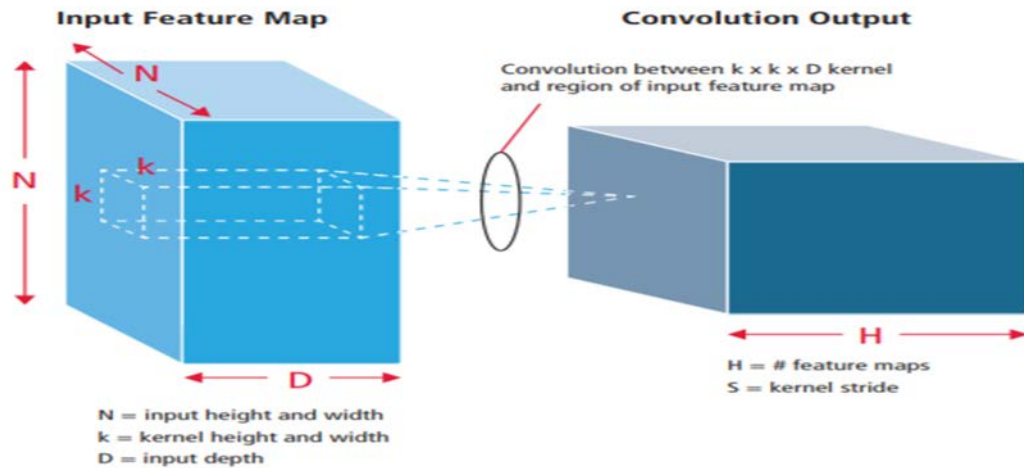
(Zhang,2018: 436)

Το μοτίβο συνδεσιμότητας μεταξύ των νευρώνων, μιμείται τη δομή του βιολογικού οπτικού φλοιού. Ο οπτικός φλοιός περιέχει πολλά κύτταρα τα οποία είναι υπεύθυνα για την ανίχνευση φωτός σε μικρές, επικαλυπτόμενες υποπεριοχές του οπτικού πεδίου (κελιά), οι οποίες ονομάζονται δεκτικά πεδία και λειτουργούν ως τοπικά φίλτρα για το δεδομένο εισόδου. Στην περίπτωση του Συνελεκτικού Νευρωνικού Δικτύου, την λειτουργία των κυττάρων στον οπτικό φλοιό, την εκτελεί το συνελεκτικό επίπεδο (Hijazi et al., 2015: 3). Ο κάθε νευρώνας του δικτύου, δρα μόνο σε ένα περιορισμένο μέρος της οπτικής περιοχής του δεδομένου εισόδου, το οποίο ορίζεται ως δεκτικό πεδίο (receptive field), τα δεκτικά αυτά πεδία, λόγω του πλήθους των νευρώνων, αλληλεπικαλύπτονται, με τέτοιο τρόπο ώστε να καλύπτεται ολόκληρη η οπτική περιοχή (LeCun et al., 1998:2283). Συνεπώς, σε αντίθεση με τα τυπικά Τεχνητά Νευρωνικά Δίκτυα, στα δίκτυα αυτά, οι νευρώνες σε οποιοδήποτε επίπεδο μπορούν να συνδεθούν μόνο σε μια μικρή περιοχή του προηγούμενου τους επιπέδου (O'Shea & Nash, 2015:4; Sakib et al., 2018:1).

Τα Συνελεκτικά Επίπεδα (convolutional layers), όπως υποδηλώνει και το όνομα τους, κατέχουν σημαντικό ρόλο στον τρόπο λειτουργίας των Συνελεκτικών Τεχνητών Δικτύων (Sakib et al., 2018:2). Είναι το πρώτο επίπεδο του δικτύου, και χρησιμοποιείται για τον εντοπισμό και την εξαγωγή χαρακτηριστικών από τα δεδομένα εισόδου (Yaseen,2018:192). Όταν εισάγουμε μια εικόνα, αυτό που βλέπει ο υπολογιστής είναι ένας πίνακας τιμών, ανάλογα με την ανάλυση και το μέγεθος της

εικόνας, ο υπολογιστής θα δει διαφορετικούς πίνακες, όπως για παράδειγμα, έναν πίνακα $32 \times 32 \times 3$ (ο αριθμός 3 αναφέρεται σε τιμές RGB – εξαρτάται από τον τύπο του δεδομένου) (Zhang, 2018: 436). Κατ' επέκταση, όταν εισάγουμε μια εικόνα στο δίκτυο, έχει μέγεθος $\langle N \times N \times D \rangle$, ταυτόχρονα στο δίκτυο, έχουμε έναν αριθμό $\langle H \rangle$ από πυρήνες (kernels), όπου ο καθένας τους έχει μέγεθος $\langle k \times k \times D \rangle$ και αποσκοπούν την εξαγωγή τοπικών χαρακτηριστικών από τα δεδομένα. Η συνέλιξη του δεδομένου εισόδου μαζί με τον αριθμό πυρήνων, παράγει έναν $\langle H \rangle$ αριθμό από χάρτες χαρακτηριστικών, δηλαδή χαρακτηριστικών που έχουν εντοπιστεί στο δεδομένο εισόδου (feature map) (Hijazi et al., 2015:4). Πιο συγκεκριμένα, μέσω των δεκτικών πεδίων, οι νευρώνες μπορούν να εξαγάγουν στοιχειώδη οπτικά και τοπικά χαρακτηριστικά, από το δεδομένο εισαγωγής. Το δεκτικό πεδίο ενός νευρώνα που σχετίζεται με συγκεκριμένη περιοχή στο προηγούμενο στρώμα, σχηματίζει έναν φορέα βάρους, ο οποίος παραμένει ίσος σε όλα τα σημεία του επιπέδου. Καθώς οι νευρώνες στο επίπεδο μοιράζονται τα ίδια βάρη, μπορούν να ανιχνευθούν τα παρόμοια χαρακτηριστικά που εμφανίζονται σε διαφορετικές τοποθεσίες στο δεδομένο εισόδου (Indolia et al., 2018:681) .

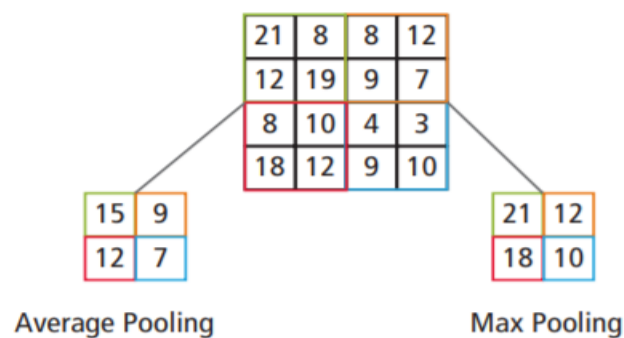
Ο πυρήνας (kernel) (ή αλλιώς φίλτρο), ολισθαίνει πάνω από το διάνυσμα εισόδου, ανιχνεύοντας χαρακτηριστικά με σκοπό τη δημιουργία ενός χάρτη χαρακτηριστικών (feature map). Αυτή η μέθοδος ολίσθησης του πυρήνα οριζόντια καθώς και κάθετα ονομάζεται λειτουργία συνέλιξης. Διαφορετικοί πυρήνες θα παράγουν διαφορετικούς χάρτες χαρακτηριστικών για την ίδια εικόνα. Αυτή η λειτουργία εξάγει έναν $\langle H \rangle$ αριθμό χαρακτηριστικών από την εικόνα εισόδου και αντιπροσωπεύει έναν $\langle H \rangle$ αριθμό πυρήνων και $\langle H \rangle$ χάρτες χαρακτηριστικών (Indolia et al., 2018:681). Κατ' επέκταση, ένα συνελικτικό επίπεδο αποτελείται από διάφορους χάρτες χαρακτηριστικών (με διαφορετικά διανύσματα βάρους), έτσι ώστε να μπορούν να εξαχθούν πολλά χαρακτηριστικά σε κάθε τοποθεσία (LeCun et al., 1998:2283).



(Hijazi et al., 2015:5)

Σχ.7: Διαδικασία συνέλιξης

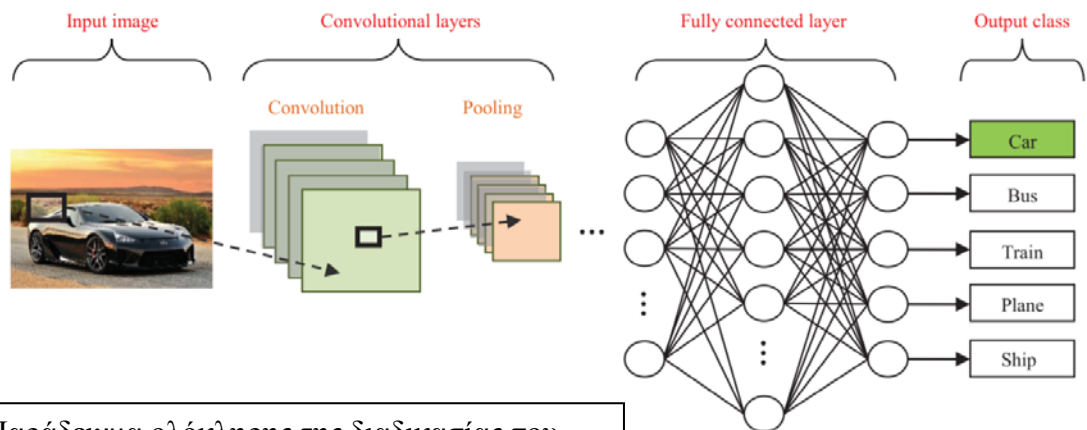
Ως συνέχεια της πορείας του συνελκτικού επιπέδου, έχουμε την είσοδο στο επίπεδο συγκέντρωσης (pooling layers). Σε ένα Συνελκτικό Νευρωνικό Δίκτυο, μπορούμε να έχουμε εναλλαγές μεταξύ των συνελκτικών επιπέδων και των επιπέδων συγκέντρωσης (μετά το συνελκτικό επίπεδο ακολουθεί επίπεδο συγκέντρωσης) (Albelwi & Mahmood, 2017:4). Σκοπός του συγκεκριμένου επιπέδου, είναι η μείωση του μεγέθους του χώρου της εικόνας και να κάνει τα χαρακτηριστικά πιο «ανθεκτικά» ως προς τον θόρυβο και την παραμόρφωση. Οι λειτουργίες συγκέντρωσης εφαρμόζονται ξεχωριστά σε κάθε χάρτη χαρακτηριστικών. Ο κύριος ρόλος του επιπέδου συγκέντρωσης είναι η δειγματοληψία, η οποία μειώνει περαιτέρω τον αριθμό των παραμέτρων αφαιρώντας ασήμαντα δείγματα στο χάρτη δυνατοτήτων (Zhang, 2018: 437; Hijazi et al., 2015:5).



(Hijazi et al., 2015:5)

Σχ. 8: Παραδείγματα διαδικασίας συγκέντρωσης

Το αμέσως επόμενο στάδιο, είναι τα Πλήρως συνδεδεμένα επίπεδα, κατά τα οποία οι νευρώνες του ενός επιπέδου συνδέονται με κάθε νευρώνα του επόμενου επιπέδου, ενώ σκοπός τους, είναι η διενέργεια ταξινόμησης. Το πλήρως συνδεδεμένο επίπεδο λαμβάνει τις εξόδους των ανωτέρων επιπέδων, δηλαδή τους χάρτες χαρακτηριστικών που έχουν δημιουργηθεί σε υψηλότερο επίπεδο, και καθορίζει ποια κατηγορία ταιριάζει καλύτερα σε αυτά τα χαρακτηριστικά (Zhang,2018: 437)



Σχ. 9: Παράδειγμα ολόκληρης της διαδικασίας που ακολουθεί ένα Συνελικτικό Νευρωνικό Δίκτυο

(Rawat & Wang, 2017: 2355)

Κεφάλαιο 3

Συστήματα Αναγνώρισης Πινακίδων Κυκλοφορίας

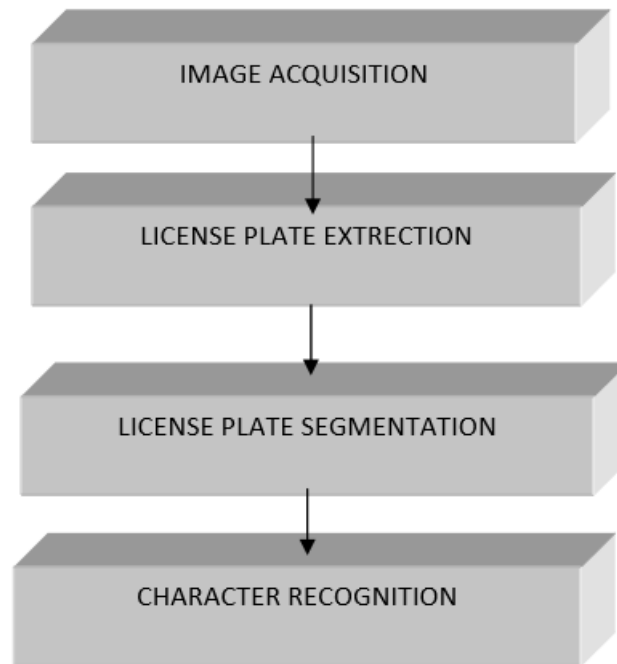
3.1 Ανάπτυξη Συστημάτων Αναγνώρισης Πινακίδων Κυκλοφορίας	18
3.2 Χαρακτηριστικά Πινακίδων Κυκλοφορίας για την περίπτωση της Κυπριακής Δημοκρατίας	22

3.1 Ανάπτυξη Συστημάτων Αναγνώρισης Πινακίδων Κυκλοφορίας

Η αυτόματη αναγνώριση πινακίδων έχει οριστεί ως η μέθοδος εξαγωγής των αλφαριθμητικών στοιχείων της πινακίδας ενός αυτοκινήτου, από μια στατική εικόνα ή από καρέ κινούμενης εικόνας, για τον σκοπό αυτόν, γίνεται χρήση τεχνολογιών επεξεργασίας εικόνας και αναγνώρισης χαρακτήρων (Optical Character Recognition – OCR) (Qadri & Asif, 2009: 335). Τα συστήματα αναγνώρισης πινακίδων κυκλοφορίας έχουν οριστεί με διάφορες ονομασίες όπως LPR (License Plate Recognition), ALPR (Automatic license plate Recognition), ANPR (Automatic Number Plate Recognition) και άλλα πολλά. Για την εκπόνηση της παρούσας εργασίας θα χρησιμοποιείται ο όρος LPR.

Η διαδικασία που συνήθως ακολουθεί ένα LPR σύστημα, ξεκινάει από την εισαγωγή της εικόνας ή καρέ από κινούμενη εικόνα στο σύστημα, ακολούθως, ο αλγόριθμος του LPR συστήματος διενεργεί τρία βασικά βήματα επεξεργασίας (Pooja & Malik,2018: 101) . Πρώτο βασικό βήμα επεξεργασίας είναι η αναγνώριση και η απομόνωση της περιοχής της εικόνας που περιλαμβάνει τους αριθμούς εγγραφής του οχήματος. Δεύτερο βασικό βήμα είναι η κατάτμηση του κάθε χαρακτήρα που περιλαμβάνεται στις πινακίδες κυκλοφορίας και τέλος, το τρίτο βασικό βήμα είναι η αναγνώριση του κάθε χαρακτήρα. Το πρώτο και το δεύτερο βήμα ενσωματώνουν τεχνικές επεξεργασίας της εικόνας / βίντεο, με σκοπό την καλύτερη αναγνώριση των χαρακτήρων και την μείωση

πιθανοτήτων σφάλματος (Anagnostopoulos et al. , 2008: 377). Η πιο πάνω διαδικασία, διαφαίνεται και στο σχήμα 10:



Σχ. 10: Διαδικασία που ακολουθεί ένα LPR σύστημα

(Maulin & Bhalodia, 2014: 99)

Το πρώτο βήμα του αλγόριθμου που χρησιμοποιούν τα LPR συστήματα είναι η αναγνώριση και η απομόνωση της περιοχής της εικόνας, στην οποία βρίσκονται οι αριθμοί εγγραφής του οχήματος. Το βήμα αυτό είναι αρκετά δύσκολο και απαιτητικό, λόγω ποικιλομορφίας των διάφορων πινακίδων καθώς και λόγω διάφορων συνθηκών που πιθανών να επηρέασαν τη λήψη της φωτογραφίας, όπως για παράδειγμα η έλλειψη επαρκούς φωτισμού (Anagnostopoulos et al. , 2008: 377). Η αναγνώριση της περιοχής της πινακίδας μπορεί να γίνει με διάφορες μεθόδους οι οποίες έχουν κατηγοριοποιηθεί και αξιοποιούνται ανάλογα των απαιτήσεων και της χρησιμότητας τους ως προς το σύστημα. Κάποιες από τις κατηγορίες των μεθόδων αυτών, χωρίζονται σε υποκατηγορίες γιατί μπορεί να χρησιμοποιούν παρόμοιες τεχνικές αλλά η κάθε μέθοδος εισάγει τις δικές της παραλλαγές για καλύτερα αποτελέσματα με βάση τις ανάγκες του συστήματος (Anagnostopoulos et al., 2008: 377).

Μέθοδοι που μπορούν να αξιοποιηθούν για την υλοποίηση του πρώτου βήματος, της αναγνώρισης δηλαδή και την απομόνωσης της περιοχής της εικόνας όπου απεικονίζεται

η πινακίδα κυκλοφορίας, είναι η δυαδική επεξεργασία εικόνας (Binary Image Processing) , επεξεργασία της εικόνας στα διάφορα επίπεδα του γκριζου (Gray-Level Processing), επεξεργασία του χρώματος της εικόνας (Color Processing), η χρήση ταξινομητών (Classifiers) (Anagnostopoulos et al. , 2008: 377-382), μεθόδους που χρησιμοποιούν κυρίως τεχνικές οι οποίες είναι βασισμένες στον συνδυασμό της στατιστικής ακμών και μαθηματικής μορφολογίας και επιτυγχάνουν αρκετά καλά αποτελέσματα (Pooja & Malik , 2018: 103) κ.α. .

Φυσικά το πρώτο βήμα του αλγόριθμου μπορεί να υλοποιηθεί με μία από τις πιο πάνω μεθόδους ή με συνδυασμό κάποιων από τις μεθόδους, όπως επίσης αυτές οι μέθοδοι μπορούν να υλοποιηθούν με την χρήση διάφορων τεχνικών ή με συνδυασμό τεχνικών. Για παράδειγμα κατά την μέθοδο του Binary Image Processing μπορεί να χρησιμοποιηθεί η τεχνική όπου υπολογίζεται η γραφική παράσταση και η μεταβολή μιας εικόνας. Η τεχνική αυτή χρησιμοποιείται έχοντας κατά νου ότι η αλλαγή της φωτεινότητας στην περιοχή όπου βρίσκονται οι αριθμοί εγγραφής του οχήματος είναι πιο ευδιάκριτη και πιο εμφανές από ότι σε άλλα σημεία. Επίσης κατά την μέθοδο της Gray-Level Processing χρησιμοποιείται η τεχνική μερικής ανάλυσης της εικόνας , όπου η εικόνα του οχήματος σαρώνεται με απόσταση N-σειράς και καταμετρώνται οι ακμές που εμφανίζονται. Εάν ο αριθμός των υπαρχόντων ακμών είναι μεγαλύτερος από μια τιμή κατωφλίου (Threshold) τότε η παρούσα τεχνική υποθέτει ότι στην περιοχή υπάρχουν οι αριθμοί εγγραφής του οχήματος (Anagnostopoulos et al. , 2008:378).

Το δεύτερο βασικό βήμα του αλγόριθμου που χρησιμοποιούν τα LPR συστήματα, είναι η κατάτμηση των χαρακτήρων του αριθμού εγγραφής του οχήματος. Στο προηγούμενο βήμα εντοπίστηκε και απομονώθηκε η περιοχή που βρίσκεται η πινακίδα με τους αριθμούς εγγραφής του οχήματος. Αφού η περιοχή της πινακίδας απομονώνεται, στόχος του δεύτερου βήματος είναι η κατάτμηση και ο διαχωρισμός των χαρακτήρων που υπάρχουν στην πινακίδα. Όπως και στο προηγούμενο βήμα υπάρχουν διάφορες τεχνικές όπου μπορούν να αξιοποιηθούν για την υλοποίηση του σταδίου αυτού και οι οποίες έχουν κατηγοριοποιηθεί με βάση την μεθοδολογία τους. Η υλοποίηση μπορεί να γίνει με την δυαδική επεξεργασία εικόνας (Binary Image Processing), επεξεργασία εικόνας στα επίπεδα του γκριζου (Gray-Level Processing) και με την χρήση ταξινομητών (Classifiers). Παρόμοια με το προηγούμενο βήμα κατά την επεξεργασία

της απομονωμένης εικόνας που απεικονίζει την πινακίδα μπορούν να χρησιμοποιηθούν διάφορες τεχνικές της μεθόδου που αξιοποιήθηκε με σκοπό την κατάτμηση των χαρακτήρων της πινακίδας (Anagnostopoulos et al. , 2008: 384).

Τέλος το τρίτο βήμα του αλγόριθμου που χρησιμοποιούν τα LPR συστήματα είναι η αναγνώριση των χαρακτήρων, οι οποίοι έχουν ήδη διαχωριστεί στο προηγούμενο βήμα του αλγορίθμου. Κάποιες από τους μεθόδους που χρησιμοποιούνται για αναγνώριση χαρακτήρων είναι με την χρήση στατιστικών ταξινομητών (Classifiers), με τεχνικές αρχιτεκτονικής υπολογιστικής νοημοσύνης (Computational intelligence architectures) όπως τα νευρωνικά δίκτυα , με την κοινή αντιστοίχιση προτύπων και με την απόσταση Hausdorff (Mahabadi & Ranjbar ,2010 : 55).

Αναμφίβολα οι αλγόριθμοι, οι μέθοδοι και οι τεχνικές που μπορούν να χρησιμοποιηθούν για την υλοποίηση ενός LPR συστήματος ποικίλουν. Στην πιο πάνω έρευνα αναλύσαμε τα κύρια βήματα που χρησιμοποιούνται συνήθως στην υλοποίηση ενός τέτοιου συστήματος και έγινε μια γενική αναφορά σε μεθόδους και τεχνικές που μπορούν να αξιοποιηθούν. Πολλές από τις μεθόδους και τις τεχνικές χαρακτηρίζονται ως περίπλοκες και απαιτούν αρκετό χρόνο μέχρι να δώσουν το αποτέλεσμα του συστήματος (Beibut & Magzhan & Chingiz, 2014:1). Για τον λόγο αυτόν, οι διάφορες μέθοδοι και τεχνικές που υπάρχουν, θα μελετηθούν και θα αξιολογηθούν με σκοπό την επιλογή των καταλληλότερων για την ανάπτυξη και υλοποίηση του LPR συστήματος με βάση των αναγκών, των απαιτήσεων και των προδιαγραφών του συστήματος μας. Οι μέθοδοι και οι τεχνικές που θα αξιοποιηθούν θα αναφερθούν και θα επεξηγηθούν σε μετέπειτα στάδιο.

3.2 Χαρακτηριστικά Πινακίδων Κυκλοφορίας για την περίπτωση της Κυπριακής Δημοκρατίας

Για τον ορθό σχεδιασμό και ανάπτυξη ενός LPR συστήματος θα πρέπει να ληφθούν υπόψη οι διάφορες μορφές καθώς και τα χαρακτηριστικά των πινακίδων κυκλοφορίας όπου υπάρχουν στο κράτος όπου θα αξιοποιηθεί το σύστημα, ώστε να γίνεται επιτυχώς ο εντοπισμός και η αναγνώριση των πινακίδων (Singh et al., 2016:94). Στην περίπτωση της παρούσας διπλωματικής εργασίας, στόχος είναι ο σχεδιασμός και η αξιοποίηση ενός LPR συστήματος το οποίο θα αξιοποιηθεί στο κράτος της Κυπριακής Δημοκρατίας, για τον λόγο αυτόν θα εστιάσουμε και θα αναφερθούμε στις μορφές και τα χαρακτηριστικά των πινακίδων κυκλοφορίας, όπως αυτά ορίζονται από τον Νόμο και τους κανονισμούς, τα οποία και θα επιδιωχθεί να ενσωματωθούν στους αλγόριθμους του συστήματος, ενισχύοντας την λειτουργικότητα και την αποτελεσματικότητα του συστήματος.

Όπως αναφέρεται στο «Οι περί Μηχανοκινήτων Οχημάτων και Τροχαίας Κινήσεως Κανονισμοί του 1984», η μορφή και τα χαρακτηριστικά που πρέπει να έχει μια πινακίδα κυκλοφορίας είναι πολύ συγκεκριμένα, διαστάσεις, χρωματισμοί και πολλά άλλα, οφείλουν να έχουν καθορισμένη μορφή.

- Γράμματα και αριθμοί:

Κάθε πινακίδα θα πρέπει να είναι ορθογώνιου σχήματος και να φέρει τα γράμματα και αριθμούς που παραχωρούνται από τον Έφορο, οι αριθμοί εγγραφής των πινακίδων μπορούν να αποτελούνται από τα γράμματα (A B E H K M N P T X Y Z) και τους αριθμούς (1 2 3 4 5 6 7 8 9 0). Κάθε γράμμα και κάθε αριθμός πρέπει να είναι όρθιος και οι διαστάσεις τους ορίζονται σε ύψος 75 χιλιοστόμετρα και πλάτος 47,5 χιλιοστόμετρα για τα γράμματα και 44,5 χιλιοστόμετρα για τους αριθμούς. Γράμματα, αριθμοί, σήματα και λοιπές ενδείξεις τίθενται στην πινακίδα ανάγλυφα, τα άκρα των γραμμάτων, αριθμών, σημάτων και των λοιπών ενδείξεων βρίσκονται υπό γωνία 45 μοιρών.

- Τοποθέτηση και Διαστάσεις πινακίδων:

Οι πινακίδες τοποθετούνται μία στο πρόσθιο και μία στο οπίσθιο μέρος του οχήματος σε όρθια θέση. Η τοποθέτηση της πινακίδας γίνεται στο κέντρο του διαμηκούς επιπέδου συμμετρίας του οχήματος και κάθετη ή σχεδόν κάθετη προς αυτήν.

Οι διαστάσεις των πινακίδων τους καθορίζονται σε τρεις περιπτώσεις:

- (α) Περίπτωση 1: Σε περίπτωση πινακίδας μιας σειράς αναγραφής των στοιχείων, οι διαστάσεις ορίζονται σε πλάτος 520 χιλιοστόμετρα και ύψος 110 χιλιοστόμετρα
- (β) περίπτωση 2: Σε περίπτωση πινακίδας δύο σειρών αναγραφής των στοιχείων, οι διαστάσεις ορίζονται σε πλάτος 340 χιλιοστόμετρα και ύψος 200 χιλιοστόμετρα.
- (γ) περίπτωση 3: για κάθε όχημα της κατηγορίας L1e και L3e, οι διαστάσεις ορίζονται σε πλάτος 240 μέχρι 280 χιλιοστόμετρα και ύψος 200 χιλιοστόμετρα, για τις οπίσθιες πινακίδες των οχημάτων αυτών οι διαστάσεις ορίζονται σε πλάτος 240 μέχρι 280 χιλιοστόμετρα και ύψος 200 χιλιοστόμετρα. Για τις οπίσθιες πινακίδες των οχημάτων της κατηγορίας L2e, L4e, L5e, L6e και L7e οι διαστάσεις ορίζονται σε πλάτος 240 μέχρι 280 χιλιοστόμετρα και ύψος 200 χιλιοστόμετρα και , ενώ για τις εμπρόσθιες πινακίδες των οχημάτων αυτών ορίζεται το πλάτος 240 μέχρι 280 και ύψος 110 χιλιοστόμετρα. Σημειώνεται ότι για τα οχήματα κατηγορίας L1e και L3e είναι δυνατόν να χρησιμοποιούνται πινακίδες των διαστάσεων που καθορίζονται στα σημεία (α) και (β).

Σε όλες τις περιπτώσεις οι ολικές εξωτερικές διαστάσεις των πινακίδων δύναται να έχουν ανοχή ± 1 χιλιοστόμετρο.

Οι διαστάσεις των γραμμάτων και αριθμών καθώς και οι αποστάσεις που πρέπει να τηρούνται μεταξύ τους και μεταξύ αυτών και των σημάτων ή των άκρων της πινακίδας, διαφαίνονται στα σχήματα 2,3 και 4.

- Διακριτικό σημείο και σφραγίδα:

Το χρώμα του αντανakλαστικού φόντου, όπου θα τοποθετείται το διακριτικό σημείο της Κυπριακής Δημοκρατίας, θα πρέπει να είναι κυανό, σύμφωνα με το χρωματομετρικό σύστημα του MUNSELL: 5,9 PB 3,4/15,1. Το διακριτικό σημείο της Κυπριακής Δημοκρατίας, που θα τοποθετείται κάτω από τους δώδεκα αστέρες πρέπει να είναι οπισθο-αντανakλαστικό και αποτελείται από τα γράμματα «CY». Τα γράμματα CY θα έχουν το ίδιο χρώμα με την αντανakλαστική επιφάνεια του μέρους της πινακίδας, πάνω στο οποίο αναγράφονται τα διακριτικά σημεία της ταυτότητας του οχήματος, εκτός των περιπτώσεων πινακίδων στις οποίες το εν λόγω μέρος δεν είναι χρώματος άσπρου ή κίτρινου, οπότεν τα γράμματα CY θα έχουν άσπρο χρώμα.

Η σφραγίδα,

(α) στην περίπτωση πινακίδας μιας σειράς αναγραφής των στοιχείων (περίπτωση 1), τοποθετείται στο μέσο της πινακίδας

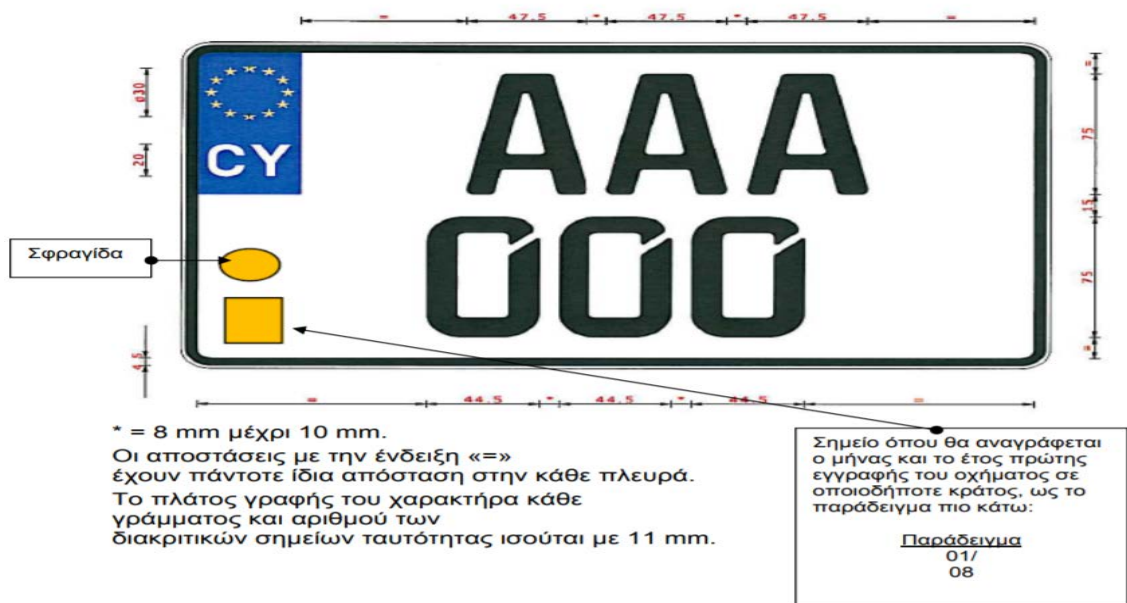
(β) στην περίπτωση πινακίδας με δύο σειρές αναγραφής των στοιχείων με διαστάσεις 240-280 x 200 (περίπτωση 2), τοποθετείται στο μέσο ενός ορθογώνιου, ύψους 112 χιλιοστωμέτρων και πλάτους 45 χιλιοστωμέτρων, που βρίσκεται στο κάτω αριστερό άκρο, μετά το περιμετρικό μαύρο πλαίσιο.

(γ) στην περίπτωση πινακίδας με δύο σειρές αναγραφής των στοιχείων με διαστάσεις 240-280 x 110 (περίπτωση 3), τοποθετείται στο μέσο της πινακίδας

Το σημείο όπου τοποθετείται η σφραγίδα και στις τρεις περιπτώσεις διαφαίνεται ξεκάθαρα στα σχήματα 2,3,4.



Σχ. 9: Δείγμα πινακίδας μιας σειράς αναγραφής των στοιχείων (520*110cm) (περίπτωση 1)



Σχ.10: Δείγμα πινακίδας διπλής σειράς αναγραφής των στοιχείων (340*200cm) (περίπτωση 2)



Σχ.11: Δείγμα πινακίδας δύο σειρών αναγραφής των στοιχείων (Περίπτωση 3)

Κεφάλαιο 4

Σχεδιασμός και Ανάπτυξη συστήματος αυτόματης αναγνώρισης πινακίδων κυκλοφορίας

4.1 Εξακρίβωση Αναγκών και Καθορισμός Απαιτήσεων	26
4.2 Καθορισμός Προδιαγραφών	27
4.3 Εργαλεία & Τεχνικές που θα χρησιμοποιηθούν	28
4.4 Σχεδιασμός και Υλοποίηση Συστήματος	31
4.5 Έλεγχος Συστήματος	42
4.6 Αποτελέσματα	44
4.7 Συσχετισμός με άλλες μελέτες	45
4.8 Δυσκολίες	47

4.1 Εξακρίβωση Αναγκών και Καθορισμός Απαιτήσεων

Κατά τον σχεδιασμό και την ανάπτυξη ενός LPR συστήματος, καλούμαστε να λάβουμε υπόψη διάφορες δυσκολίες και περιορισμούς, όπως για παράδειγμα, οι νομοθεσίες και οι κανονισμοί, οι κλιματολογικές και γεωγραφικές συνθήκες και πολλά άλλα.

Αρχικά, λόγω του διαφορετικού νομικού και κανονιστικού πλαισίου της κάθε χώρας, οι πινακίδες κυκλοφορίας των οχημάτων της κάθε χώρας, ενδέχεται να έχουν διαφορετική μορφή και χαρακτηριστικά. Κατά συνέπεια, ένα LRP σύστημα, είναι πολύ δύσκολο να αναπτυχθεί και να μπορεί να αξιοποιηθεί με μεγάλη αποτελεσματικότητα, σε οποιαδήποτε χώρα. Για τον λόγο αυτόν, εφόσον το LPR σύστημα όπου θα σχεδιάσουμε και θα αναπτύξουμε, στοχεύει στην αξιοποίηση του στην Κυπριακή Δημοκρατία, ο σχεδιασμός του θα γίνει με βάση την μορφή και τα χαρακτηριστικά που ορίζει ο νόμος και οι κανονισμοί της Κυπριακής Δημοκρατίας.

Επιπρόσθετα, ο εντοπισμός μιας πινακίδας κυκλοφορίας εντός της φωτογραφίας και η απομόνωση της από το φόντο, θεωρείται από τα κρίσιμα βήματα ενός LPR συστήματος καθώς είναι το σημείο το οποίο μπορεί να επηρεάσει περισσότερο, την αποτελεσματικότητα, την ακρίβεια καθώς και την ταχύτητα του όλου συστήματος. Οι δυσκολίες του σταδίου αυτού, προέρχονται από προβλήματα όπως η κακή ποιότητα εικόνας, η παραμόρφωση της εικόνας, η αντανάκλαση καθώς και η ομοιότητα χρωμάτων μεταξύ της πινακίδας με άλλα αντικείμενα που υπάρχουν στο φόντο, στοιχεία τα οποία δυσκολεύουν και καθυστερούν τον εντοπισμό της πινακίδας (Mohindra & Shukla, 2013: 1). Την ποιότητα φωτογραφίας των πινακίδων κυκλοφορίας και συνεπώς την αναγνωρισιμότητα και την αποτελεσματικότητα του συστήματος επηρεάζουν άμεσα και οι κλιματολογικές και γεωγραφικές συνθήκες. Πιο συγκεκριμένα, συνθήκες οι οποίες μπορούν να επηρεάσουν την λήψη και την ποιότητας της φωτογραφίας, είναι η υπερβολική φωτεινότητα, το σκοτάδι, η γωνιά του ηλιακού φωτός, η ομίχλη, η σκόνη, η υγρασία, η βροχή, το χιόνι και άλλα. Πιθανός τρόπος αντιμετώπισης των προβλημάτων αυτών είναι η χρήση καμερών υψηλής ευκρίνειας, πράγμα που σε πολλές περιπτώσεις της καθημερινότητας μας δεν είναι εφικτό, για τον λόγο αυτόν, κατά την ανάπτυξη του LPR συστήματος μας, θα αξιοποιηθούν τεχνικές και μέθοδοι που δίδουν υψηλά ποσοστά αναγνωρισιμότητας και αποτελεσματικότητας.

4.2 Καθορισμός Προδιαγραφών

Σκοπός μας είναι η εισαγωγή εικόνας από τον χρήστη εντός του συστήματος, και το σύστημα θα πρέπει να εντοπίσει εντός της εικόνας το σημείο όπου απεικονίζεται η πινακίδα εγγραφής του οχήματος. Με τον εντοπισμό του σημείου, το σύστημα θα αποκόπτει την πινακίδα εγγραφής και θα πρέπει με την να αναγνωρίσει τους χαρακτήρες που την αποτελούν. Μετά την αναγνώριση των χαρακτήρων, το σύστημα θα πρέπει να επιστρέψει στον χρήστη τους αριθμούς εγγραφής του οχήματος καθώς και τα στοιχεία του κατόχου. Το σύστημα αυτό έχει σκοπό την αξιοποίηση του εντός της Κυπριακής Δημοκρατίας, συνεπώς, το σύστημα θα εστιάζει στον εντοπισμό και την αναγνώριση πινακίδων κυκλοφορίας οχημάτων της Κύπρου.

4.3 Εργαλεία & Τεχνικές που θα χρησιμοποιηθούν

Για την ανάπτυξη του συστήματος αναγνώρισης πινακίδων κυκλοφορίας θα αξιοποιηθούν τα ακόλουθα εργαλεία και τεχνικές:

- **Σύστημα Anaconda και Γλώσσα προγραμματισμού Python**

Γίνεται χρήση του συστήματος Anaconda και αξιοποιείται η γλώσσα προγραμματισμού Python, καθώς είναι μια γλώσσα υψηλού επιπέδου ενώ ταυτόχρονα, η σχεδιαστική του φιλοσοφία, δίνει έμφαση στην αναγνωσιμότητα του κώδικα και η σύνταξή του επιτρέπει στους προγραμματιστές να εκφράζουν έννοιες σε λιγότερες γραμμές κώδικα από ό, τι θα ήταν δυνατό σε άλλες γλώσσες (Srinath, 2017: 354).

- **OpenCV**

Αξιοποιείται η OpenCV (Open Source Computer Vision Library), η οποία είναι μια βιβλιοθήκη ανοικτού λογισμικού και η οποία εμπεριέχει συναρτήσεις που αφορούν την μηχανική όραση και είναι κατάλληλες για διάφορες εφαρμογές οπτικής επεξεργασίας.

- **QtDesigner**

Το QtDesigner είναι ένα εργαλείο σχεδίασης και δημιουργίας γραφικών περιβάλλοντων διεπαφής (GUIs – Graphical User Interfaces).

- **Darknet**

Το Darknet είναι ένα πλαίσιο νευρωνικού δικτύου, ανοιχτού κώδικα, γραμμένο σε C και CUDA. Είναι γρήγορο, εύκολο στην εγκατάσταση και υποστηρίζει υπολογισμούς CPU και GPU. Το Darknet, προσφέρεται μέσω του Github, από τον AlexeyAB.

- **OIDv4 (Open Images Dataset V4)**

Το OIDv4, είναι ένα σύνολο δεδομένων εικόνων (9,2M) με ενοποιημένους σχολιασμούς για ταξινόμηση εικόνας, ανίχνευση αντικειμένων και ανίχνευση οπτικών σχέσεων (Kuznetsova et al., 2020: 1). Η εγκατάσταση του έγινε μέσω του Github.

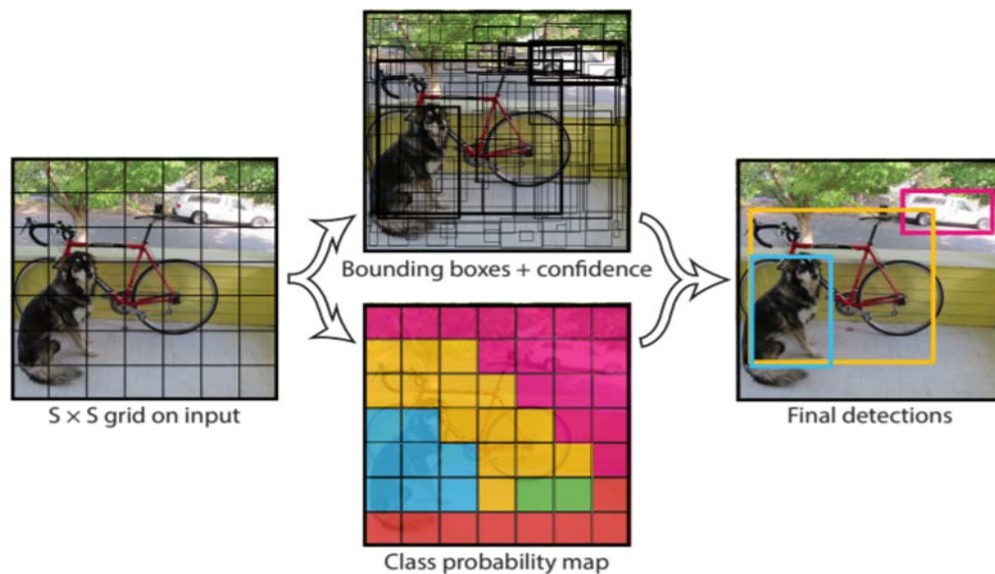
- **Αλγόριθμος YOLOv3**

Ο αλγόριθμος YOLO (You Only Look Once) αποτελεί έναν σύγχρονο αλγόριθμο μονού σταδίου που αποσκοπεί στον εντοπισμό αντικειμένων σε εικόνα ή βίντεο, σε μία μόνο εκτέλεση του αλγορίθμου βάσει ενός Συνελεκτικού Νευρωνικού Δικτύου (βασίζεται στο δίκτυο Darknet-53, το οποίο διαθέτει ένα δίκτυο 53 επιπέδων). Ο αλγόριθμος αυτός σε αντίθεση με άλλες περιπτώσεις, για τον εντοπισμό του αντικειμένου, δεν εξετάζει ολόκληρη την εικόνα, αλλά τμήματα της εικόνας τα οποία έχουν μεγάλες πιθανότητες να περιέχουν το αντικείμενο. Συνεπώς, το YOLO λειτουργεί βασισμένο στην τεχνική της παλινδρόμησης, παίρνει δηλαδή μια εικόνα εισόδου, προβλέπει τις πιθανότητες των κλάσεων και τις συντεταγμένες του πλαισίου οριοθέτησης (Redmon, 2016: 779). Επιλέχθηκε η αξιοποίηση του αλγορίθμου YOLOv3 για τον λόγο ότι λειτουργεί πολύ πιο γρήγορα από άλλες μεθόδους ανίχνευσης (Redmon & Farhadi, 2018: 1)

Ο τρόπος με τον οποίο λειτουργεί είναι ο εξής:

1. Ο αλγόριθμος YOLO διαιρεί την εικόνα εισόδου σε πλέγμα «S x S»
2. Κάθε κελί του πλέγματος προβλέπει «B» οριοθετούμενα πλαίσια (bounding boxes) και έναν βαθμό «εμπιστοσύνης» (confidence score) για τα πλαίσια αυτά μέσω λογιστικής παλινδρόμησης, οι βαθμοί εμπιστοσύνης αντικατοπτρίζουν την βεβαιότητα και την ακρίβεια για το ότι στο πλαίσιο εμπεριέχεται αντικείμενο (αντικειμενικότητα), εάν δεν υπάρχει αντικείμενο σε αυτό το κελί, οι βαθμολογίες εμπιστοσύνης είναι μηδενικές

3. Ακολούθως, κάθε κελί του πλέγματος προβλέπει τις πιθανότητες κλάσης του αντικειμένου (The conditional class probability), είναι η πιθανότητα ότι το ανιχνευόμενο αντικείμενο ανήκει σε μια συγκεκριμένη κατηγορία
4. Στη συνέχεια πολλαπλασιάζουμε τις πιθανότητες κλάσης με τις μεμονωμένες προβλέψεις εμπιστοσύνης του κελιού και μας δίνουν βαθμολογίες εμπιστοσύνης για κάθε οριοθετημένο πλαίσιο. Μετρά δηλαδή την «εμπιστοσύνη» τόσο στην ταξινόμηση όσο και στον εντοπισμό του αντικειμένου



Σχ. 12: Η διαδικασία που ακολουθεί ο αλγόριθμος YOLO

(Redmon, 2016: 779)

- **Tesseract OCR**

Η Tesseract OCR, θεωρείται μια από τις καλύτερες μηχανές ανοικτού κώδικα (Open Source), οπτικής αναγνώρισης χαρακτήρων (Optical Character Recognition). Η Tesseract OCR, συνήθως ακολουθεί μια σειρά βημάτων, το πρώτο βήμα είναι το «Adaptive Thresholding», κατά το οποίο μετατρέπεται η εικόνα σε δυαδικές εικόνες. Ακολούθως γίνεται ανάλυση των συνδεδεμένων στοιχείων (Connected Component Analysis) κατά την οποία γίνεται εξαγωγή των περιγραμμάτων των χαρακτήρων που

εμπεριέχονται στην εικόνα, σε αυτό το στάδιο, τα περιγράμματα συγκεντρώνονται και ενσωματώνονται σε ‘blobs’. Τα ‘blobs’ είναι οργανωμένα σε γραμμές κειμένου, οι γραμμές και οι περιοχές αναλύονται για τον εντοπισμό σταθερού βήματος ή αναλογικού κειμένου. Οι γραμμές κειμένου χωρίζονται σε λέξεις με διαφορετικό τρόπο, ανάλογα με το είδος του χαρακτήρα διαχωρισμού. Ακολουθώς, η διαδικασία αναγνώρισης γίνεται μέσω διαδικασίας δύο περασμάτων (Smith, 2007:1).

4.4 Σχεδιασμός και Υλοποίηση Συστήματος

- **Εγκατάσταση προγραμμάτων και εργαλείων:**

Πρωταρχικό στάδιο για την έναρξη υλοποίησης του συστήματος, ήταν οι εγκατάσταση όλων των απαραίτητων προγραμμάτων και εργαλείων που θα αξιοποιηθούν. Αρχικά, εγκαταστάθηκε το σύστημα Anaconda, στο οποίο δημιουργήθηκε ένα περιβάλλον και στο οποίο εγκαταστάθηκαν διάφορα packages – modules, όπως το pyqt, OpenCv, pytesseract κ.α., τα οποία θα βοηθούσαν για την υλοποίηση του συστήματος.

Επιπλέον, έγινε εγκατάσταση του Daknet-53, του αλγόριθμου YOLOv3 καθώς και των CUDA και cuDNN, τα οποία είναι βιβλιοθήκες ανεπτυγμένες από την εταιρεία NVIDIA με σκοπό να βοηθήσουν τους προγραμματιστές να “τρέξουν” προγράμματα σε κάρτες γραφικών (GPU) της NVIDIA.

- **Training**

Με την χρήση του εργαλείου OVIDv4, μας επιτράπηκε να κατεβάσουμε ένα σύνολο εικόνων από το storage.googleapis.com το οποίο περιλαμβάνει διάφορα Datasets εικόνων. Για την δημιουργία του δικού μας Dataset κατεβάσαμε από την προαναφερθέν ιστοσελίδα 600 εικόνες που περιλαμβάνουν πινακίδες εγγραφής αυτοκινήτων. Με την περαιτέρω χρήση του εργαλείου OVIDv4 υπολογίστηκαν και τα annotation της κάθε εικόνας τα οποία αποθηκεύτηκαν στο Dataset μας, με την ονομασία της εικόνας. Μετέπειτα, δημιουργήσαμε το αρχείο train.txt, το οποίο περιλαμβάνει το path της κάθε εικόνας σε αλφαβητική σειρά και περιλαμβάνεται στο Dataset που δημιουργήσαμε.

Ακολούθως, δημιουργήσαμε το αρχείο MyYolov3.cfg, το οποίο είναι ένα configuration file και το οποίο χρησιμοποιείται τόσο στην εκπαίδευση όσο και στο testing/validation των αποτελεσμάτων, ανάλογα με τις αλλαγές που κάνουμε στο file. Επίσης το εν λόγω configuration file θα χρησιμοποιηθεί και σε μετέπειτα στάδιο, καθώς είναι απαραίτητο για την λειτουργία του συστήματος που θα υλοποιήσουμε. Στην ουσία κατά την ανάπτυξη του συστήματος μας, το προαναφερθέν cfg file και το weights file το οποίο θα είναι προϊόν της εκπαίδευσης, θα χρησιμοποιείται για να γίνει Load το network, δηλαδή το δίκτυο που χρησιμοποιούμε, όπου στην προκειμένη περίπτωση είναι το Darknet.

Μετέπειτα, δημιουργήσαμε το αρχείο classes.names το οποίο αφορά τα ονόματα των κλάσεων / αντικειμένων, τα οποία θα εκπαιδευτεί ώστε να εντοπίζει το σύστημα μας. Στην παρούσα περίπτωση, το αρχείο αυτό περιλαμβάνει μόνο το όνομα «Vehicle registration plate», καθώς το σύστημα μας θα εκπαιδευτεί στον εντοπισμό πινακίδων εγγραφής αυτοκινήτων.

Επιπρόσθετα, για την εκπαίδευση χρειαζόμασταν και το αρχείο darknet53.conv.74 το οποίο προσφέρεται από το ίδιο το DarkNet και περιέχει τα weights, το ίδιο το δίκτυο. Ουσιαστικά, είναι ένα pre-trainer feature Extractor και χρησιμοποιείται σαν η «ραχοκοκαλιά» του αλγόριθμου YOLOv3.

Στη συνέχεια, τα αρχεία που δημιουργήθηκαν τοποθετήθηκαν στον φάκελο στον οποίο έγινε build το DarkNet στον υπολογιστή μας. Όταν πλέον είχαμε όλα τα απαραίτητα αρχεία και τα τοποθετήσαμε στους φακέλους που θέλαμε, δημιουργήσαμε το αρχείο Dataset.data. Το εν λόγω αρχείο, περιλαμβάνει αρχικά πόσες κλάσεις αφορά η εκπαίδευση ή το testing/validation. Ακολούθως στο αρχείο περιλαμβάνεται το path του train.txt που δημιουργήσαμε προηγουμένως όπως και του test.txt το οποίο χρησιμοποιείται στο στάδιο του testing. Επίσης, περιέχει το path του αρχείου classes.names καθώς και το path στο οποίο θα αποθηκευτεί το αρχείο weights που θα προκύψει από την εκπαίδευση του dataset μας.

Εν κατακλείδι, μέσω του terminal και της εντολής «darknet_no_gpu detector train data/datasets.data cfg/myyolov3.cfg darknet53.conv.7», όπου μετά από πολύωρη εκπαίδευση του δικτύου προέκυψε το αρχείο MyWeights.weights..

```
371: 1.371572, 1.453708 avg loss, 0.000019 rate, 1111.860000 seconds, 23744 images, 1226.834413 hours left
Loaded: 0.000000 seconds
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 82 Avg (IOU: 0.802335, GIOU: 0.797248), Class: 0.416144, Obj: 0.007618,
No Obj: 0.003024, .5R: 1.000000, .75R: 1.000000, count: 1, class_loss = 0.669275, iou_loss = 0.050696, total_loss = 0.719970
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 94 Avg (IOU: 0.000000, GIOU: 0.000000), Class: 0.000000, Obj: 0.000000,
No Obj: 0.001499, .5R: 0.000000, .75R: 0.000000, count: 1, class_loss = 0.006990, iou_loss = 0.000000, total_loss = 0.006990
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 106 Avg (IOU: 0.353679, GIOU: 0.197628), Class: 0.671251, Obj: 0.004715,
No Obj: 0.000755, .5R: 0.000000, .75R: 0.000000, count: 2, class_loss = 1.113137, iou_loss = 1.540434, total_loss = 2.653571
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 82 Avg (IOU: 0.000000, GIOU: 0.000000), Class: 0.000000, Obj: 0.000000,
No Obj: 0.002975, .5R: 0.000000, .75R: 0.000000, count: 1, class_loss = 0.005882, iou_loss = 0.000000, total_loss = 0.005882
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 94 Avg (IOU: 0.000000, GIOU: 0.000000), Class: 0.000000, Obj: 0.000000,
No Obj: 0.001496, .5R: 0.000000, .75R: 0.000000, count: 1, class_loss = 0.006957, iou_loss = 0.000000, total_loss = 0.006957
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 106 Avg (IOU: 0.596300, GIOU: 0.562692), Class: 0.457565, Obj: 0.001076,
No Obj: 0.000733, .5R: 1.000000, .75R: 0.000000, count: 2, class_loss = 1.303180, iou_loss = 0.620569, total_loss = 1.923749
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 82 Avg (IOU: 0.000000, GIOU: 0.000000), Class: 0.000000, Obj: 0.000000,
No Obj: 0.002975, .5R: 0.000000, .75R: 0.000000, count: 1, class_loss = 0.006164, iou_loss = 0.000000, total_loss = 0.006164
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 94 Avg (IOU: 0.418853, GIOU: 0.403367), Class: 0.437387, Obj: 0.001385,
No Obj: 0.001437, .5R: 0.500000, .75R: 0.000000, count: 2, class_loss = 1.321251, iou_loss = 1.533311, total_loss = 2.854562
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 106 Avg (IOU: 0.167398, GIOU: -0.326537), Class: 0.688247, Obj: 0.000202,
No Obj: 0.000735, .5R: 0.000000, .75R: 0.000000, count: 1, class_loss = 0.556632, iou_loss = 3.780105, total_loss = 4.336737
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 82 Avg (IOU: 0.000000, GIOU: 0.000000), Class: 0.000000, Obj: 0.000000,
No Obj: 0.002937, .5R: 0.000000, .75R: 0.000000, count: 1, class_loss = 0.005937, iou_loss = 0.000000, total_loss = 0.005937
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 94 Avg (IOU: 0.451111, GIOU: 0.449293), Class: 0.480590, Obj: 0.001510,
No Obj: 0.001448, .5R: 0.000000, .75R: 0.000000, count: 1, class_loss = 0.639802, iou_loss = 0.743546, total_loss = 1.383347
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 106 Avg (IOU: 0.149422, GIOU: -0.010117), Class: 0.640579, Obj: 0.001779,
No Obj: 0.000734, .5R: 0.000000, .75R: 0.000000, count: 1, class_loss = 0.570920, iou_loss = 2.911708, total_loss = 3.482627
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 82 Avg (IOU: 0.000000, GIOU: 0.000000), Class: 0.000000, Obj: 0.000000,
No Obj: 0.003029, .5R: 0.000000, .75R: 0.000000, count: 1, class_loss = 0.006082, iou_loss = 0.000000, total_loss = 0.006082
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 94 Avg (IOU: 0.000000, GIOU: 0.000000), Class: 0.000000, Obj: 0.000000,
No Obj: 0.001557, .5R: 0.000000, .75R: 0.000000, count: 1, class_loss = 0.007413, iou_loss = 0.000000, total_loss = 0.007413
```

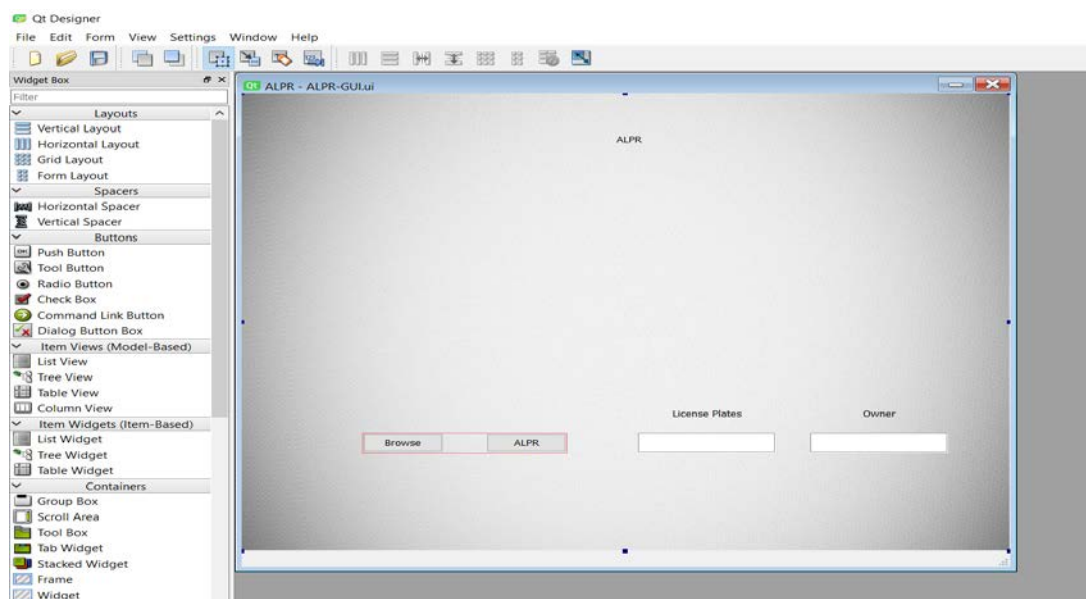
• Δημιουργία του GUI

Για την δημιουργία του GUI του συστήματος, χρησιμοποιήσαμε το QtDesigner. Με το φιλικό προς τον χρήστη περιβάλλον που προσφέρεται από το QtDesigner, η δημιουργία περιβάλλοντος διεπαφής του χρήστη με το σύστημα, ήταν μια αρκετά απλή διαδικασία.

Για το main window, χρησιμοποιήσαμε ένα Qwidget στο οποίο περιέχονται δύο Qpushbuttons, έξι Qlabels και δύο QTextEdits. Το πρώτο QLabel είναι ένα κενό label το οποίο περιλαμβάνεται σαν container για το background. Τα επόμενα δύο QLabels που είναι επίσης κενά, χρησιμοποιούνται όπως και το προαναφερθέν, σαν containers για να μπορούμε να παρουσιάσουμε τις εικόνες που θέλουμε στο mainWindow, ένα Qlabel για ολόκληρη την φωτογραφία που θα επιλέξει ο χρήστης και ένα Qlabel για την φωτογραφία που δημιουργείται από το σύστημα που θα περιλαμβάνει μόνο την περιοχή που βρίσκεται η πινακίδα εγγραφής του οχήματος. Τα υπόλοιπα τρία Qlabels δεν είναι κενά, στο ένα αναγράφεται «ALPR» και είναι τοποθετημένο στο πάνω μέρος, στην

μέση του main window και λειτουργεί σαν τίτλος. Τα άλλα δύο QLabel, βρίσκονται πάνω από τα δύο QTextEdits και στο πρώτο αναγράφεται «License Plates» και στο δεύτερο αναγράφεται «Owner» αφού στα QTextEdits που βρίσκονται από κάτω τους, θα εμφανιστούν οι αριθμοί εγγραφής και ο ιδιοκτήτης του υπό μελέτη οχήματος αντίστοιχα. Το πρώτο pushbutton αφορά το «Browse», που όταν επιλεγθεί από τον χρήστη, του επιτρέπει να επιλέξει την φωτογραφία του οχήματος από οποιοδήποτε location είναι αποθηκευμένη στον υπολογιστή του. Με την επιλογή της φωτογραφίας, η φωτογραφία εμφανίζεται στο main window πάνω από τα δύο pushbuttons στο ένα από τα δύο κενά QLabel που περιγράψαμε πιο πάνω. Το δεύτερο pushbutton αφορά το «ALPR», που στην ουσία όταν επιλεγθεί από τον χρήστη, θέτει σε λειτουργία το σύστημα. Με το πάτημα του «ALPR», στο δεύτερο κενό QLabel του main window, εμφανίζεται η φωτογραφία που έχει δημιουργηθεί και η οποία περιλαμβάνει την πινακίδα με τους αριθμούς εγγραφής του οχήματος. Στα δύο QTextEdits εμφανίζονται οι αριθμοί εγγραφής και ο ιδιοκτήτης του υπό μελέτη οχήματος.

Όταν τελειώσαμε το QtDesigner δημιούργησε ένα αρχείο UI. Για την μετατροπή του UI αυτού σε γλώσσα προγραμματισμού Python, χρησιμοποιήσαμε το τερματικό που προσφέρεται από το anaconda. Έγινε activate το περιβάλλον που δημιουργήσαμε στο anaconda καθώς περιλάμβανε το pyqt package. Τέλος μέσω της εντολής `pyuic5 -x Alpr-Gui.ui -o Alpr.py` δημιουργήσαμε το `Alpr.py`.



```

class Ui_AnprWindow(object):
    def setupUi(self, AnprWindow):

        AnprWindow.setObjectName("AnprWindow")
        AnprWindow.resize(890, 695)

        self.centralwidget = QtWidgets.QWidget(AnprWindow)
        self.centralwidget.setObjectName("centralwidget")

        #Background
        self.Background = QtWidgets.QLabel(self.centralwidget)
        self.Background.setGeometry(QtCore.QRect(0, 0, 891, 701))
        self.Background.setText("")
        self.Background.setPixmap(QtGui.QPixmap("Background.jpg"))
        self.Background.setObjectName("Background")

        # Text Holder for Owner
        self.TextOw = QtWidgets.QTextEdit(self.centralwidget)
        self.TextOw.setGeometry(QtCore.QRect(690, 520, 160, 30))
        sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Fixed, QtWidgets.QSizePolicy.Fixed)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)
        sizePolicy.setHeightForWidth(self.TextOw.sizePolicy().hasHeightForWidth())
        self.TextOw.setSizePolicy(sizePolicy)
        self.TextOw setFrameShape(QtWidgets.QFrame.Box)
        self.TextOw setFrameShadow(QtWidgets.QFrame.Sunken)
        self.TextOw.setTextInteractionFlags(QtCore.Qt.NoTextInteraction)
        self.TextOw.setObjectName("TextOw")

```

- **Εισαγωγή Φωτογραφίας**

Για την εισαγωγή της φωτογραφίας από τον χρήστη δημιουργήσαμε ένα function με το όνομα `browse_image`, το οποίο ενώσαμε με το `QPushButton` «Browse».

```

def retranslateUi(self, AnprWindow):
    _translate = QtCore.QCoreApplication.translate
    AnprWindow.setWindowTitle(_translate("AnprWindow", "ALPR"))
    self.LicensePlatesLabel.setText(_translate("AnprWindow", "License Plates"))
    self.OwnerLabel.setText(_translate("AnprWindow", "Owner"))
    self.Browse.setText(_translate("AnprWindow", "Browse"))
    self.Browse.clicked.connect(self.browse_image)
    self.Alpr.setText(_translate("AnprWindow", "ALPR"))
    self.Alpr.clicked.connect(self.Alpr_image)
    self.Title.setText(_translate("AnprWindow", "ALPR"))

def browse_image(self):
    global image_path
    filename = QFileDialog.getOpenFileName()
    image_path = filename[0]
    pixmap = QtGui.QPixmap(image_path)
    pixmap = pixmap.scaled(350, 350, QtCore.Qt.KeepAspectRatio)
    self.Image.setPixmap(pixmap)

```

Με το πιο πάνω function η φωτογραφία που επιλέγεται από τον χρήστη παρουσιάζεται στο main window, στο κενό `QLabel` με την ονομασία «Image».

- **Εντοπισμός και απομόνωση της περιοχής της πινακίδα εγγραφής**

Για την διαδικασία του εντοπισμού και της απομόνωσης της περιοχής που βρίσκεται η πινακίδα εγγραφής του οχήματος, όπως αναφέραμε και σε προηγούμενο στάδιο, χρησιμοποιήσαμε τον αλγόριθμο YOLOv3. Για την υλοποίηση του αλγορίθμου, αρχικά έπρεπε να γίνει αρχικοποίηση των παραμέτρων `confThreshold` , `nmsTheshold` , `InpWidth`, `InpHeight` τα οποία αφορούν το confidence threshold , το non-maximum suppression threshold, το width και το height της φωτογραφίας που θα εισάγουμε στο network αντίστοιχα.

```
# Initialize the parameters
confThreshold = 0.5 #Confidence threshold
nmsThreshold = 0.4 #Non-maximum suppression threshold

inpWidth = 608      #Width of network's input image
inpHeight = 608     #Height of network's input image
```

Στην συνέχεια έπρεπε να φορτώσουμε στο network τα αρχεία που χρειάζονται και τα οποία δημιουργήσαμε κατά το στάδιο του training πιο πάνω.

```
# Load names of classes
classesFile = "classes/classes.names";

classes = None
with open(classesFile, 'rt') as f:
    classes = f.read().rstrip('\n').split('\n')

# Give the configuration and weight files
Configuration = "cfg/Myyolov3.cfg";
Weights = "weights/Myweights.weights";

# Load the network using the weights and the configurations.
net = cv.dnn.readNetFromDarknet(Configuration, Weights)
net.setPreferableBackend(cv.dnn.DNN_BACKEND_OPENCV)
net.setPreferableTarget(cv.dnn.DNN_TARGET_CPU)
```

Ακολουθώς, δημιουργήσαμε ένα function με το όνομα `Alpr_image` στο οποίο, ενώσαμε το `Qpushbutton` «ALPR». Αρχικά μέσω του `OpenCv` ανοίγουμε την φωτογραφία που επέλεξε ο χρήστης και δημιουργούμε ένα `blob` της φωτογραφίας, με μέγεθος το `inpWidth` και το `inpHeight` που αρχικοποιήσαμε προηγουμένως. Στην συνέχεια, χρησιμοποιούμε το `blob` σαν είσοδο στο δίκτυο μας. Από το δίκτυο παίρνουμε τα αποτελέσματα στην μεταβλητή `outs`, στην οποία περιλαμβάνονται τα διάφορα `bounding boxes` που δημιούργησε το δίκτυο βάσει των `weights` με τα οποία το φορτώσαμε. Ακολουθώς, ελέγχουμε τα αποτελέσματα που μας επιστραφήκαν και κρατούμε τα `bounding boxes` που έχουν `confidence` μεγαλύτερο από το `confidence threshold` που θέσαμε αρχικά. Στην συνέχεια, χρησιμοποιούμε το `non-maximum suppression Threshold` για να εξαλείψουμε όλα τα `overlapping bounding boxes` που δημιούργησε το δίκτυο.

```
frame = cv.imread(image_path)
blob = cv.dnn.blobFromImage(frame, 1 / 255, (inpWidth, inpHeight), [0, 0, 0], 1, crop=False)
net.setInput(blob)
outs = net.forward(getOutputsNames(net))
frameHeight = frame.shape[0]
frameWidth = frame.shape[1]
#Check all the bounding boxes created from the network and keep those that has confidence > confidence Threshold
classIds = []
confidences = []
boxes = []
for out in outs:
    for detection in out:
        scores = detection[5:]
        classId = np.argmax(scores)
        confidence = scores[classId]
        if confidence > confThreshold:
            center_x = int(detection[0] * frameWidth)
            center_y = int(detection[1] * frameHeight)
            width = int(detection[2] * frameWidth)
            height = int(detection[3] * frameHeight)
            left = int(center_x - width / 2)
            top = int(center_y - height / 2)
            classIds.append(classId)
            confidences.append(float(confidence))
            boxes.append([left, top, width, height])
```

```
# Use non-maximum suppression Threshold, to eliminate The overlapping Bounding Boxes
indices = cv.dnn.NMSBoxes(boxes, confidences, confThreshold, nmsThreshold)
for i in indices:
    i = i[0]
    box = boxes[i]
    left = box[0]
    top = box[1]
    width = box[2]
    height = box[3]
    cv.rectangle(frame, (left, top), (left + width, top + height), (0, 255, 0), 3)
    LicensePlate = imgCreator(frame, top, left, height, width)
    pixmap = QtGui.QPixmap("License_Plate.jpg")
    pixmap = pixmap.scaled(200, 200, QtCore.Qt.KeepAspectRatio)
```

Ακολουθώντας, δημιουργήσαμε μια συνάρτηση με το όνομα `ImgCreator` η οποία βασισμένη στο `bounding box` που έχει δημιουργηθεί στην εικόνα μας, δημιουργεί μια νέα εικόνα, η οποία απομονώνει την περιοχή που βρίσκεται η πινακίδα εγγραφής του οχήματος. Μετέπειτα, γίνεται `resize` και αποθηκεύεται στον υπολογιστή μας με το όνομα `License_Plate.jpg`. Στην συνέχεια, με την χρήση του δεύτερου κενού `QLabel`, εμφανίζεται η νέα εικόνα, η οποία απεικονίζει την πινακίδα, στην προκαθορισμένη περιοχή του `main Window`.

```
#Based on a Bounding Box, crops and create the new image
def imgCreator(frame, top, left, height, width):
    LicensePlate = frame[top:top + height, left:left + width]
    width_cropped = int(LicensePlate.shape[1] * 250 / 100)
    height_cropped = int(LicensePlate.shape[0] * 250 / 100)
    dsize = (width_cropped, height_cropped)
    LicensePlate = cv.resize(LicensePlate, dsize)
    cv.imwrite('License_plate.jpg', LicensePlate)
    return LicensePlate
```

- **Αναγνώριση Αριθμών Εγγραφής**

Το στάδιο αυτό, αφορά την εξαγωγή των αριθμών εγγραφής από την περιοχή που βρίσκεται η πινακίδα εγγραφής. Για την αναγνώριση των αριθμών της πινακίδας εγγραφής, χρησιμοποιήσαμε το OCR `tesseract`. Το `tesseract` για να χρησιμοποιηθεί, χρειάζεται αρχικά να δηλώσουμε το `root folder` που βρίσκεται το `tesseract.exe`.

```
#Tesseract root Folder
pytesseract.pytesseract.tesseract_cmd = r"C:\Users\an7re\AppData\Local\Tesseract-OCR\tesseract.exe"
```

Στην συνέχεια, δημιουργήσαμε μια συνάρτηση με το όνομα `filteringImage`, η οποία παίρνει σαν παράμετρο την εικόνα που περιλαμβάνει την περιοχή με την πινακίδα εγγραφής που δημιουργήσαμε στο προηγούμενο βήμα. Μέσω της συνάρτησης, επεξεργαζόμαστε την εικόνα, προσθέτοντας της διάφορα φίλτρα, όπως να την μετατρέψουμε σε `grayscale` ή να γίνει `deskew`, ούτως ώστε να ενισχύσουμε την αναγνώριση των χαρακτήρων της πινακίδας.

```
def FilteringImage(LicensePlate):
    imgfilter = cv.cvtColor(LicensePlate, cv.COLOR_RGB2GRAY)
    imgfilter = cv.threshold(imgfilter, 0, 255, cv.THRESH_BINARY | cv.THRESH_OTSU)[1]
    imgfilter = cv.GaussianBlur(imgfilter, (5, 5), 0)
    kernel = np.ones((5, 5), np.uint8)
    imgfilter = cv.erode(imgfilter, kernel, iterations=1)
    imgfilter = cv.dilate(imgfilter, kernel, iterations=1)
    coords = np.column_stack(np.where(imgfilter > 0))
    angle = cv.minAreaRect(coords)[-1]

    if angle < -45:
        angle = -(90 + angle)
    else:
        angle = -angle
    (h, w) = imgfilter.shape[:2]
    center = (w // 2, h // 2)
    M = cv.getRotationMatrix2D(center, angle, 1.0)
    imgfilter = cv.warpAffine(imgfilter, M, (w, h), flags=cv.INTER_CUBIC, borderMode=cv.BORDER_REPLICATE)
    return imgfilter
```

Στην συνέχεια, επεξεργαζόμαστε το configuration του tesseract, μετά από δοκιμές επιλέξαμε το `-psm 6`, γιατί είχαμε καλύτερα αποτελέσματα. Το `psm` αφορά το `pagesegmode` με το οποίο θα εργαστεί το tesseract.

```
pagesegmode values are:
0 = Orientation and script detection (OSD) only.
1 = Automatic page segmentation with OSD.
2 = Automatic page segmentation, but no OSD, or OCR
3 = Fully automatic page segmentation, but no OSD. (Default)
4 = Assume a single column of text of variable sizes.
5 = Assume a single uniform block of vertically aligned text.
6 = Assume a single uniform block of text.
7 = Treat the image as a single text line.
8 = Treat the image as a single word.
9 = Treat the image as a single word in a circle.
10 = Treat the image as a single character.
```

Ακολουθώς χρησιμοποιούμε την συνάρτηση του `tesseract.image_to_string` με παραμέτρους, την εικόνα που επεξεργαστήκαμε και το configuration στο οποίο επιλέγουμε το `-psm 6`. Η συνάρτηση, θα μας επιστρέψει ένα string με τους αριθμούς εγγραφής των πινακίδων.

```
LPstring = pytesseract.image_to_string(imgfilter, config=config)
```

Λόγω του ότι το σύστημα αναγνώριζε στην πινακίδα κάποιους μη επιθυμητούς χαρακτήρες οι οποίοι δεν εμπίπτουν στα γράμματα και τους αριθμούς που απαρτίζουν τους αριθμούς εγγραφής, δημιουργήσαμε την συνάρτηση EnhanceString. Η συνάρτηση δέχεται σαν είσοδο ένα string το οποίο επεξεργάζεται απαλείφοντας τους ανεπιθύμητους χαρακτήρες. Για την εξάλειψη των χαρακτήρων και την επεξεργασία του string, λάβαμε υπόψη μας την μορφή των κυπριακών πινακίδων εγγραφής.

```
def EnhanceString(String):
    NumberPlates = ""
    UnwantedChars = ["a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q","r","s","t","u","v","w","x","y","z"]

    LetterString = ""
    NumberString = ""

    countLetters = 0
    countNumbers = 0

    for character in String:
        if character.isalnum():
            NumberPlates += character

    for character in UnwantedChars:
        NumberPlates=NumberPlates.replace(character,"")

    NumberPlates = NumberPlates.replace('I', '')
    midlen = len(NumberPlates) / 2

    if (len(NumberPlates) > 6):
        for character in NumberPlates:
            if character.isalpha():
                countLetters = countLetters + 1
            else:
                countNumbers = countNumbers + 1
        if (countLetters > 3):
            NumberPlates = NumberPlates[1:]
        elif (countNumbers > 3):
            NumberPlates = NumberPlates[:int(midlen)] + NumberPlates[int(midlen) + 1:]

    LetterString, NumberString = NumberPlates[:int(len(NumberPlates) / 2)], NumberPlates[int(len(NumberPlates) / 2):]
    NumberString = NumberString.replace('G', '0')
    NumberString = NumberString.replace('U', '0')
    return LetterString + " " + NumberString
```

Οι αριθμοί εγγραφής εμφανίζονται στον χρήστη στο QTextEdit που βρίσκεται στο main window.

- **Εξαγωγή δεδομένων από βάση δεδομένων**

Στην συνέχεια, δημιουργήσαμε μια συνάρτηση με το όνομα MatchOwner. Η συνάρτηση αυτή, δέχεται σαν είσοδο το string που δημιουργήσαμε στο προηγούμενο βήμα και αφορά τους αριθμούς εγγραφής του οχήματος. Η συνάρτηση ενώνεται με την «φανταστική» βάση που δημιουργήσαμε και βάσει των αριθμών εγγραφής του οχήματος, εντοπίζουμε τον ιδιοκτήτη του οχήματος.

```

def MatchOwner(LP):

    LP=LP.replace(" ","")
    #database connection
    connection = pymysql.connect("127.0.0.1", "root", "1234", "anpr",3308)
    cursor = connection.cursor()

    Name=""
    Surname=""

    retrieve = 'SELECT * FROM cars '

    # executing the quires
    cursor.execute(retrieve)
    rows = cursor.fetchall()
    for row in rows:
        if (row[3]==LP):
            Name=row[1]
            Surname=row[2]

    #committing the connection then closing it.
    connection.commit()
    connection.close()
    return Name+" "+Surname

```

Server: MySQL 3308 » Database: anpr » Table: cars

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐ 1	ID	int(11)			No	None			Change Drop More
☐ 2	Name	varchar(10)	utf8_general_ci		No	None			Change Drop More
☐ 3	Surname	varchar(15)	utf8_general_ci		No	None			Change Drop More
☐ 4	Plate	varchar(6)	utf8_general_ci		No	None			Change Drop More
☐ 5	Model	enum('BMW','MERCEDES','AUDI','PEUGEOT','MAZDA'...	utf8_general_ci		No	None			Change Drop More

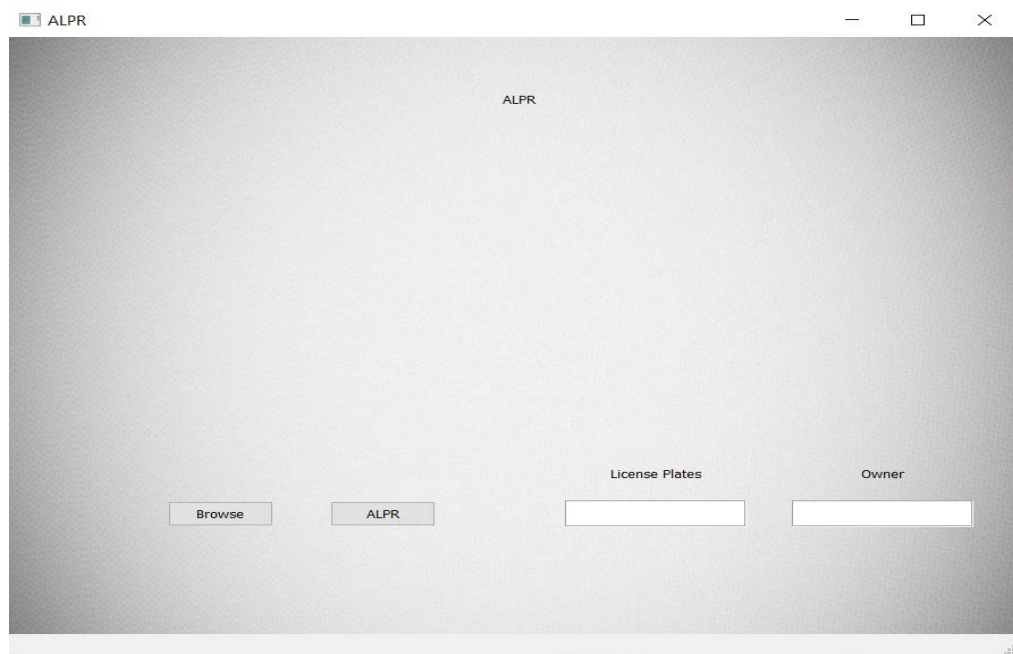
☐ Check all With selected: Browse Change Drop Primary Unique Index Fulltext Fulltext

Print Propose table structure Move columns Normalize

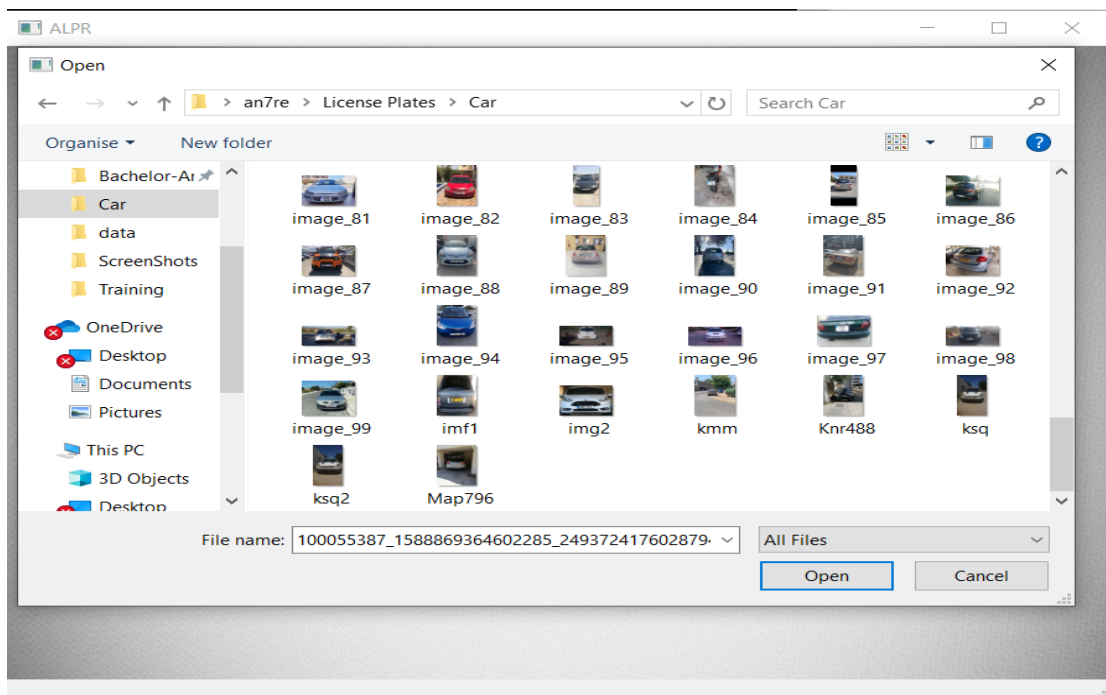
Add 1 column(s) after Model Go

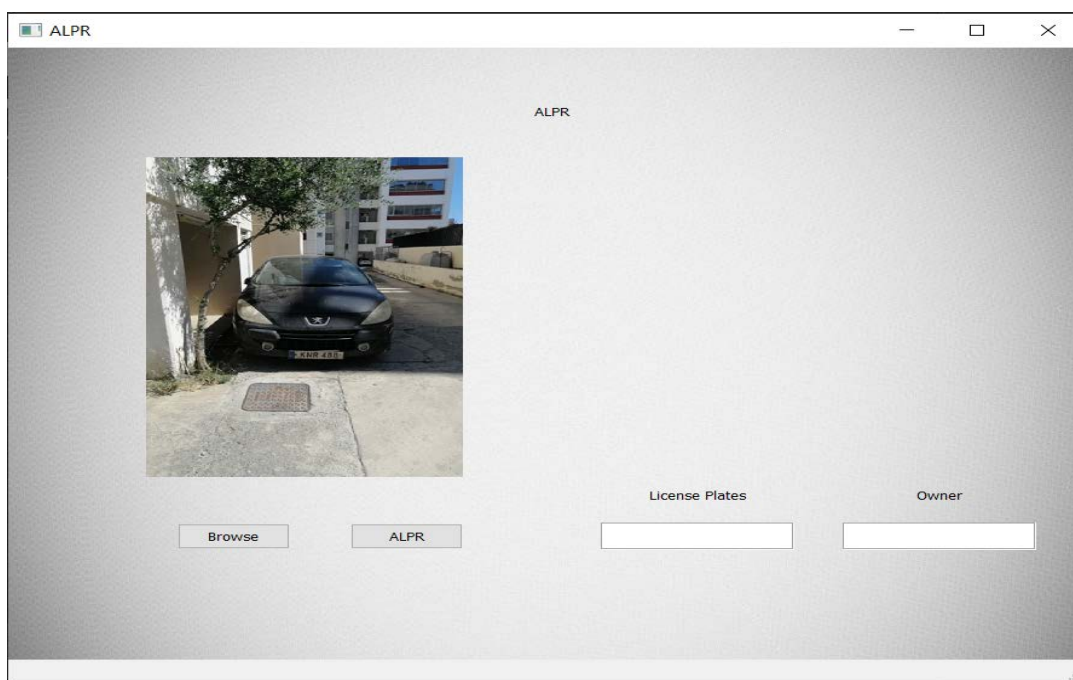
4.5 Έλεγχος Συστήματος

Η αρχική σελίδα του συστήματος μας, έχει την εξής μορφή για τον χρήστη:

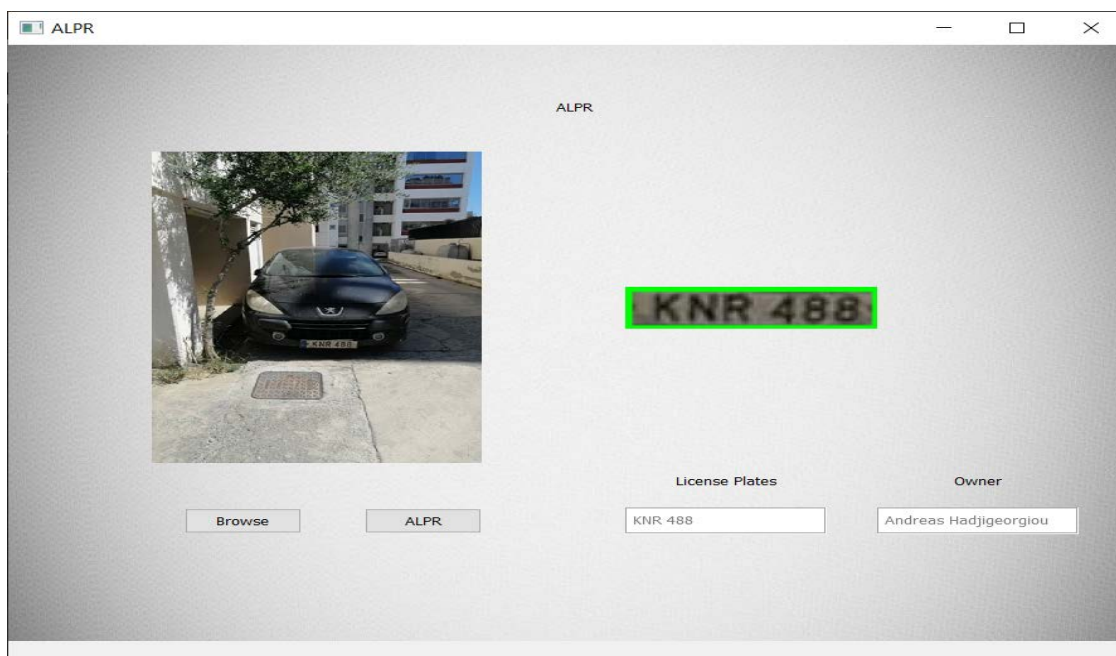


Πατώντας το κουμπί «Browse», ο χρήστης μπορεί να επιλέξει την φωτογραφία οχήματος για την οποία θέλει να γίνει αναγνώριση της πινακίδας κυκλοφορίας.





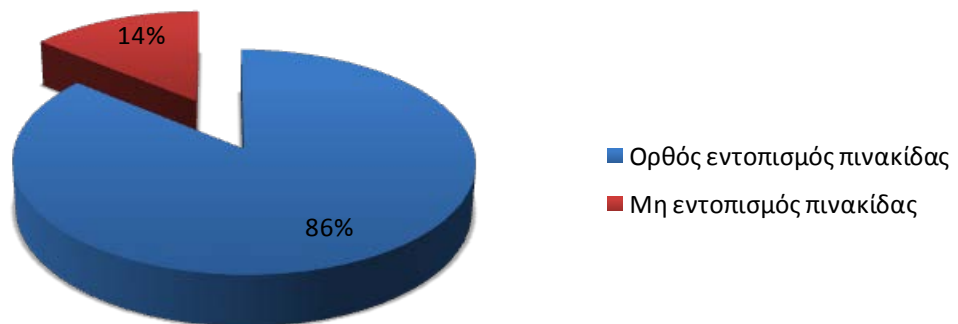
Αφού ο χρήστης επιλέξει την φωτογραφία του οχήματος, πατάει το κουμπί «ALPR», για να γίνει η αναγνώριση της πινακίδας κυκλοφορίας και το σύστημα επιστρέφει στον χρήστη τον αριθμό εγγραφής του οχήματος καθώς και το Ονοματεπώνυμο του κατόχου.



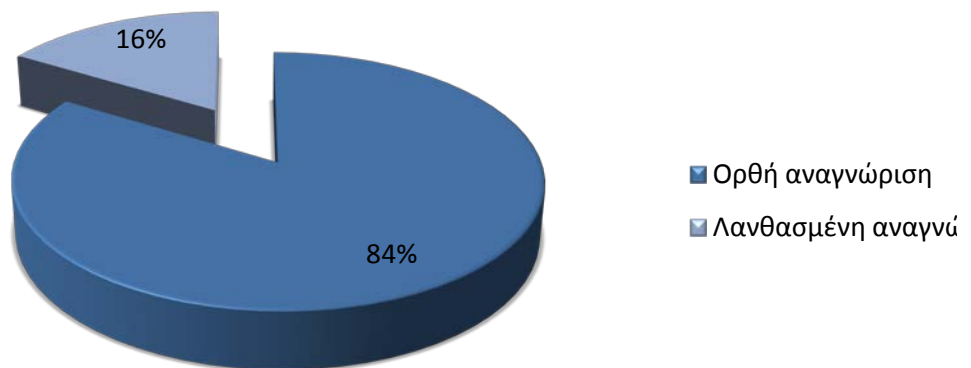
4.6 Αποτελέσματα

Για τον έλεγχο της λειτουργικότητας και της αποτελεσματικότητας του συστήματος μας, χρησιμοποιήσαμε φωτογραφίες από 150 αυτοκίνητα. Από τα 150 αυτοκίνητα έγινε ορθός εντοπισμός πινακίδας σε 129 αυτοκίνητα. Από τα 129 αυτοκίνητα, έγινε ορθή αναγνώριση των αριθμών εγγραφής σε 108 πινακίδες κυκλοφορίας.

Εντοπισμός πινακίδας κυκλοφορίας



Αναγνώριση αριθμών εγγραφής



Το κυριότερο πρόβλημα το οποίο εντοπίστηκε και το οποίο επηρέασε τον εντοπισμό της πινακίδας κυκλοφορίας, είναι η ποιότητα της φωτογραφίας. Πιο συγκεκριμένα, στις περιπτώσεις όπου η ευκρίνεια της φωτογραφίας δεν είναι καλή, επηρεάζονται τα χαρακτηριστικά της πινακίδας, με αποτέλεσμα να δυσχεραίνει την διαδικασία εντοπισμού της. Για τον ίδιο λόγο, παρατηρήθηκε ότι όσο πιο μακριά είναι τοποθετημένο το όχημα εντός της φωτογραφίας, τόσο πιο δύσκολος είναι ο εντοπισμός της πινακίδας.

Επιπρόσθετα, διαπιστώθηκε ότι σημαντικό ρόλο έχει και η κλίση που έχει η πινακίδα εντός της φωτογραφίας, στις περιπτώσεις δηλαδή όπου η πινακίδα έχει κάποιον βαθμό κλίσης, το σύστημα «δυσκολεύεται» να την εντοπίσει. Είναι αξιοσημείωτο το γεγονός, ότι το σύστημα σε καμία περίπτωση δεν εντόπισε κάποιο άλλο στοιχείο που υπήρχε στην φωτογραφία, ως πινακίδα κυκλοφορίας.

Στην περίπτωση της αναγνώρισης των αριθμών εγγραφής των πινακίδων κυκλοφορίας, παρουσιάστηκε και πάλι το πρόβλημα της ποιότητας της εικόνας, καθώς επηρεάζεται η ποιότητα των περιγραμμάτων των χαρακτήρων. Επιπρόσθετα, τα συνηθέστερα λάθη που παρατηρήθηκαν κατά την αναγνώριση, ήταν η σύγχυση μεταξύ των “G” “0” και “Q”, γεγονός που μπορούμε να το αποδώσουμε στην ομοιότητα των ψηφίων αυτών.

4.7 Συσχετισμός με άλλες μελέτες

Σε μια προσπάθεια συσχετισμού και γενίκευσης τόσο των αποτελεσμάτων του συστήματος που έχουμε αναπτύξει, όσο και των αποτελεσμάτων άλλων συστημάτων που έχουν αναπτυχθεί ώστε να εξυπηρετήσουν τον ίδιο σκοπό, διαφαίνεται ότι μέχρι στιγμής, δεν έχει επιτευχθεί η «απόλυτη» αποτελεσματικότητα ενός τέτοιου είδους συστήματος, αλλά είμαστε πολύ κοντά σε αυτό. Επιπρόσθετα, διαφάνηκε ότι στις πλείστες περιπτώσεις, οι παράγοντες οι οποίοι επηρεάζουν την αποτελεσματικότητα των συστημάτων, είναι οι ίδιοι.

Κάποια παραδείγματα συστημάτων που έχουν αναπτυχθεί μέχρι στιγμής, είναι των Chang et. Al (2004), κατά το οποίο, αξιοποιήθηκε ένας συνδυασμός τεχνικών

εντοπισμού της πινακίδας και το SO OCR για την αναγνώριση των αριθμών εγγραφής, το σύστημα τους, επέφερε 98% ακρίβεια κατά τον εντοπισμό των πινακίδων κυκλοφορίας και 95% ακρίβεια κατά την αναγνώριση των αριθμών εγγραφής. Το σύστημα που ανέπτυξε ο Sajjad (2010), στο οποίο αξιοποιήθηκε το OpenCV, επέφερε 92% ακρίβεια κατά τον εντοπισμό των πινακίδων κυκλοφορίας και 94% ακρίβεια κατά την αναγνώριση των αριθμών κυκλοφορίας. Κατά την περίπτωση των Agbemenu & Yankey & Addo (2018), οι οποίοι αξιοποίησαν για την ανάπτυξη του συστήματος τους, το OpenCV και το Tesseract OCR, επέφεραν 60% ακρίβεια κατά τον εντοπισμό των πινακίδων κυκλοφορίας και 87% ακρίβεια κατά την αναγνώριση των πινακίδων κυκλοφορίας. Στην περίπτωση του συστήματος που ανέπτυξαν οι Hendry & Chen (2019), κατά το οποίο αξιοποιήθηκε ο αλγόριθμος YOLO, το σύστημα επέφερε 98% ακρίβεια κατά τον εντοπισμό των πινακίδων κυκλοφορίας και 78% ακρίβεια κατά την αναγνώριση των πινακίδων κυκλοφορίας. Επιπλέον, στην περίπτωση των Pinto et. al (2019), όπου αξιοποιείται ο αλγόριθμος YOLOv3 σε όλα τα στάδια ανάπτυξης του συστήματος, επέφερε 95% ακρίβεια κατά τον εντοπισμό των πινακίδων κυκλοφορίας και 87% ακρίβεια κατά την αναγνώριση των πινακίδων κυκλοφορίας, στην περίπτωση αυτή, η απώλεια αποτελεσματικότητας από την αναγνώριση των αριθμών εγγραφής, οφειλόταν στον μη επιτυχή εντοπισμό των ψηφίων «I» και «1».

Οι διάφορες μέθοδοι και τεχνικές που αξιοποιούνται για την ανάπτυξη ενός LPR συστήματος, σε συνδυασμό με διάφορους παράγοντες και συνθήκες κατά τις οποίες δοκιμάζεται και αξιοποιείται το κάθε σύστημα, μπορούν να επηρεάσουν σε μεγάλο βαθμό την αποτελεσματικότητα του συστήματος. Πιο συγκεκριμένα, τόσο η απόσταση από την οποία γίνεται η λήψη, όσο και η ποιότητα της φωτογραφίας του οχήματος που θα αξιοποιηθεί στο σύστημα, έχουν άμεση επιρροή στην αποτελεσματικότητα του συστήματος, όσο πιο καλή είναι η ποιότητα και η ανάλυση της φωτογραφίας τόσο πιο καλύτερα θα είναι και τα αποτελέσματα του συστήματος. Επιπρόσθετα, πλείστα από τα σφάλματα που προκύπτουν κατά των εντοπισμό των πινακίδων αλλά και την αναγνώριση των αριθμών εγγραφής, προκύπτουν από την κλίση της πινακίδας εντός της φωτογραφίας και την προσπάθεια ευθυγράμμισης της μέσω επεξεργασίας. Επιπλέον, η ποικιλομορφία που μπορούν να έχουν οι χαρακτήρες που αποτελούν τις πινακίδες κυκλοφορίας, μπορούν να έχουν άμεσο αντίκτυπο στο στάδιο της αναγνώρισης των αριθμών εγγραφής καθώς μπορεί να μην αναγνωριστεί κάποιο ψηφίο

είτε να υπάρξει μπέρδεμα, ανάμεσα σε παρόμοιας μορφής ψηφία (Ahmad, 2015: 639-640).

4.8 Δυσκολίες

Κατά τον σχεδιασμό και την υλοποίηση του συστήματος μου, ήρθαμε αντιμέτωποι με διάφορες δυσκολίες. Αρχικά, διαπιστώθηκε η ανάγκη για υπολογιστή πιο υψηλής υπολογιστικής δύναμης, έτσι ώστε να υποστηρίξει τον όγκο των διάφορων διεργασιών που χρειάζονταν να γίνουν για την δημιουργία του συστήματος.

Επιπλέον, η εκπαίδευση του αλγορίθμου ήταν μια αρκετά χρονοβόρα διαδικασία, γεγονός που εμπόδιζε την δυνατότητα εκπαίδευσης του με μεγαλύτερο όγκο δεδομένων, όπου θα ενίσχυε την αποτελεσματικότητα του συστήματος.

Επιπρόσθετα, λόγω της πρόσφατης αλλαγής της νομοθεσίας σχετικά με την μορφή και τα χαρακτηριστικά των πινακίδων κυκλοφορίας και λόγω του ότι στο παρόν στάδιο δεν πληρούν τα συγκεκριμένα χαρακτηριστικά (του καινούριου νόμου) όλες οι υπάρχοντες πινακίδες κυκλοφορίας της Κύπρου, χρειάστηκε να υπάρχει μεγάλη ευελιξία ως προς την μορφή και τα χαρακτηριστικά όπου θα έπρεπε να αναγνωρίζει το σύστημα, με αποτέλεσμα την μείωση της αποτελεσματικότητας του.

Κεφάλαιο 5

Συμπεράσματα και Μελλοντική εργασία

5.1 Συμπεράσματα	48
5.2 Μελλοντική εργασία	49

5.1 Συμπεράσματα

Συμπερασματικά, το σύστημα που έχουμε δημιουργήσει έχει αρκετά καλά ποσοστά λειτουργικότητας και αποτελεσματικότητας και εξυπηρετεί σε μεγάλο βαθμό τον σκοπό για τον οποίο έχει σχεδιαστεί. Ως κυριότερος παράγοντας ο οποίος επηρεάζει την αποτελεσματικότητα του συστήματος, έχει καθοριστεί η ποιότητα της φωτογραφίας που εισάγει ο χρήστης.

Βάσει των δυσκολιών που αντιμετωπίσαμε καθώς και των προβλημάτων του συστήματος, εντοπίσαμε στοιχεία του συστήματος όπου με κάποιες διαφοροποιήσεις θα μπορέσουν να αυξήσουν την αποτελεσματικότητα του συστήματος μας. Μέσω των διαφοροποιήσεων αυτών, οι οποίες αναφέρονται πιο κάτω, το σύστημα μας, θα μπορέσει να αξιοποιηθεί με μεγάλη επιτυχία σε πολλούς τομείς της Κυπριακής Δημοκρατίας, κυρίως όμως στα Σώματα Ασφαλείας.

Είναι σαφές, ότι ένα σύστημα αναγνώρισης πινακίδων κυκλοφορίας, μπορεί να αξιοποιηθεί σε πολλούς τομείς της καθημερινότητας μας, και η ανάγκη πλέον, για τέτοιου είδους συστήματα, είναι επιτακτική. Θεωρούμε εύλογο ότι, ανάλογα με τον τομέα στον οποίο θα αξιοποιηθεί το σύστημα, μπορούν να γίνουν και οι ανάλογες διαφοροποιήσεις και τροποποιήσεις, ώστε να εξυπηρετεί τις ξεχωριστές ανάγκες της κάθε περίπτωσης.

5.2 Μελλοντική εργασία

Κάποιες προτεινόμενες εισηγήσεις για μελλοντικές αναβαθμίσεις του συστήματος:

Από τους αρχικούς στόχους για την ανάπτυξη του συστήματος, ήταν να εντάσσεται στο σύστημα ένα βίντεο με κίνηση αυτοκινήτων και το σύστημα να δημιουργεί καρέ εικόνων, έτσι ώστε να εντοπίζει όλες τις πινακίδες αυτοκινήτων που εμφανίζονται στο βίντεο. Λόγω της ανάγκης υψηλών υπολογιστικών συστημάτων για την διεκπεραίωση αυτής της διαδικασίας, ο στόχος αυτός δεν κατάφερε να υλοποιηθεί στο παρόν στάδιο. Σε μεταγενέστερο όμως στάδιο, με την χρήση υψηλών υπολογιστικών συστημάτων, μπορεί να προστεθεί η χρήση και επεξεργασία βίντεο.

Επιπρόσθετα, ακόμη ένας αρχικός στόχος κατά την ανάπτυξη του συστήματος, ήταν να λειτουργεί βάσει των προκαθορισμένων χαρακτηριστικών πινακίδων κυκλοφορίας όπως ορίζονται από τον νόμο. Λόγω όμως της πρόσφατης αλλαγής της νομοθεσίας, δεν ήταν δυνατό το πρόγραμμα να εστιάζει μόνο σε αυτά τα χαρακτηριστικά, καθώς με αυτόν τον τρόπο, οι πινακίδες που δεν έχουν αντικατασταθεί μέχρι στιγμής, δεν θα μπορούσαν να εντοπιστούν. Κατ' επέκταση, σε μελλοντικό στάδιο, το σύστημα μπορεί να αναπροσαρμοστεί, έτσι ώστε να εστιάζει στα συγκεκριμένα χαρακτηριστικά ώστε να αυξηθεί το ποσοστό ορθού εντοπισμού της πινακίδας και αναγνώρισης των αριθμών εγγραφής της.

Όπως διαφάνηκε, καθοριστικό ρόλο για την αποτελεσματικότητα του συστήματος έχει η ποιότητα της εικόνας, για τον λόγο αυτόν θα ήταν σημαντικό να χρησιμοποιούνται κάμερες υψηλής ευκρίνειας. Επιπλέον, μπορεί να γίνει σύνδεση του συστήματος με IP cameras, έτσι ώστε να γίνεται real time αναγνώριση των πινακίδων, χωρίς να χρειάζεται να παρέχει τα δεδομένα στο σύστημα ο χρήστης.

Επιπλέον, θα μπορούσε να γίνει εκπαίδευση του YOLOv3 μόνο με εικόνες στις οποίες να υπάρχουν πινακίδες κυκλοφορίας της Κυπριακής Δημοκρατίας, θα είναι απαραίτητο όμως να χρησιμοποιηθεί ένας μεγάλος αριθμός τέτοιων εικόνων. Επίσης, θα μπορούσε να γίνει εκπαίδευση του Tesseract OCR, έτσι ώστε να ανταποκρίνεται καλύτερα στην δομή των πινακίδων κυκλοφορίας, της Κυπριακής Δημοκρατίας.

Επιπρόσθετα, μια ακόμη αναβάθμιση η οποία μπορεί να γίνει στο σύστημα, είναι η σύνδεση του με βάσεις δεδομένων με αληθινά στοιχεία, ώστε να μπορεί να αξιοποιηθεί σε περιπτώσεις της καθημερινότητας μας.

Τέλος, θα μπορούσε για το σύστημα αυτό να δημιουργηθεί μια εφαρμογή, η οποία μέσω της χρήσης της cameras της συσκευής όπου είναι εγκατεστημένη η εφαρμογή, θα μπορεί να κάνει αυτόματη αναγνώριση των πινακίδων κυκλοφορίας και να παρέχει στον χρήστη πληροφορίες για τον κάτοχο και για το όχημα.

Βιβλιογραφία

Ελληνική:

Γεωργούλη, Α., 2015. *Τεχνητή νοημοσύνη: Μια εισαγωγική προσέγγιση* [ηλεκτρ. βιβλ.] Αθήνα:Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Διαθέσιμο στο: <http://hdl.handle.net/11419/3381>

Κ.Δ.Π.66/1984 «Οι περί Μηχανοκινήτων Οχημάτων και Τροχαίας Κινήσεως Κανονισμοί του 1984»

Πλέρου, Α. (2012). «Τεχνητά νευρωνικά δίκτυα προσομοίωσης του ανθρώπινου εγκεφάλου». *Ανοικτή Εκπαίδευση: το περιοδικό για την Ανοικτή και εξ Αποστάσεως Εκπαίδευση και την Εκπαιδευτική Τεχνολογία*, 8(1), 128-135.
doi:<https://doi.org/10.12681/jode.9794>

Αγγλική:

Agbemenu, A. S., Yankey, J., Addo, E.O. (2018). “An Automatic Number Plate Recognition System using OpenCV and Tesseract OCR Engine”. *International Journal of Computer Applications*, 180 (43):1-5. DOI: 10.5120/ijca2018917150

Ahmad, I. S., Boufama, B., Hbashi, P., Anderson, W., Elamsy, T. (2015). “Automatic License Plate Recognition: A Comparative Study”. 2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT) , Abu Dhabi, pp. 635-640, doi: 10.1109/ISSPIT.2015.7394415.

Albelwi, S., Mahmood, A.. (2017). “A Framework for Designing the Architectures of Deep Convolutional Neural Networks”. *Entropy*. 19,242 . DOI: 10.3390/e19060242

Anagnostopoulos, C., Anagnostopoulos, I., Psoroulas, I., Loumos, V., Kayafas, E. (2008). "License Plate Recognition From Still Images and Video Sequences: A Survey", IEEE: Transactions on intelligent transportation systems, Vol. 9, No. 3, 377:391

Arel, I., Rose, D., Karnowski, T. (2010). "Deep Machine Learning - A New Frontier in Artificial Intelligence Research [Research Frontier]", IEEE Comp. Int. Mag.. 5. 13-18.

Beibut, A., Magzhan, K., Chingiz, K. (2014). "Effective algorithms and methods for automatic number plate recognition". IEEE 8th International Conference on Application of Information and Communication Technologies (AICT), 1-4. doi:10.1109/ICAICT.2014.7035951.

Chahal, A., Gulia, P. (2019). "Machine Learning and Deep Learning", International Journal of Innovative Technology and Exploring Engineering (IJITEE). Vol. 8. DOI: 10.35940/ijitee.L3550.1081219

Chang, S. L., Chen, L. S., Chung, Y. C., Chen, S. W. (2004). "Automatic License Plate Recognition". IEEE Transactions on Intelligent Transportation Systems, 5(1), 42-53. <https://doi.org/10.1109/TITS.2004.825086>

Draghici, S. (1997). "A neural network based artificial vision system for license plate recognition", International journal of neural systems, 8 1, 113-126

Hendry, . Chen, R. C. (2019). "Automatic License Plate Recognition Via Sliding-Window Darknet-Yolo Deep Learning". Image and Vision Computing, 87. pp:47-56 <https://doi.org/10.1016/j.imavis.2019.04.007>

Hijazi, S.L., Kumar, R., Rowen, C. (2015). "Using Convolutional Neural Networks for Image Recognition". At https://ip.cadence.com/uploads/901/cnn_wp-pdf

- Indolia, S., Goswami, A., Mishra, S.P., Asopa, Pooja. (2018). "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach". *Procedia Computer Science*. 132. 679-688. <https://doi.org/10.1016/j.procs.2018.05.069>
- Kamilaris, A., Prenafeta-Boldú, F. X. (2018). "A review of the use of convolutional neural networks in agriculture", *The Journal of Agricultural Science* 1–11. <https://doi.org/10.1017/S0021859618000436>
- Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J.R., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Mallocci, M., Duerig, T., Ferrari, V. (2020). "The Open Images Dataset V4". *International Journal of Computer Vision*, 1 - 26. <https://doi.org/10.1007/s11263-020-01316-z>
- LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, 86(11), 2278-2323. <https://doi.org/10.1109/5.726791>
- Mahabadi, A., Ranjbar, A. (2010). "Reliable license plate recognition". *The CSI Journal on Computer Science and Engineering*, Vol. 7, No. 2 & 4 (b), 51-67
- Maulin, M., Bhalodia, D. (2014). "Survey on License Plate Recognition (LPR) System". *International Journal of Innovative Research in Technology*, Vol. 1, pp. 99-105
- Mijwil, M., Esen, A., Alsaadi, A. (2019). "Overview of Neural Networks". 1. 2.
- Mohindra, S., Shukla, D.S. (2013). "Plate Recognition (LPR) system for Indian Vehicle License Plate Extraction and Character Segmentation", *International Journal of Engineering Sciences & Research Technology*.
- O'Shea, K., Nash, R. (2015). "An Introduction to Convolutional Neural Networks". *ArXiv*, abs/1511.08458.

Pinto, P.M., Busson, A.J., Melo, J.P., Colcher, S., & Milidiú, R.L. (2019). "PVBR-Recog: A YOLOv3-based Brazilian Automatic License Plate Recognition Tool ". DOI: 10.5753/webmedia_estendido.2019.8149

Pooja, Malik, T. (2018). "License plate localization method based on vertical edge analysis using SVM technique". International Journal of Advance Research, Ideas and Innovations in Technology, 4(5) www.IJARIIT.com.

Qadri, M. T., Asif, M. (2009). "Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition". International Conference on Education Technology and Computer, pp.335-338. doi: 10.1109/ICAICT.2014.7035951

Rawat, W., Wang, Z . (2017). "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review". Neural computation, 29(9), 2352–2449. https://doi.org/10.1162/NECO_a_00990

Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A. (2016). "You Only Look Once: Unified, Real-Time Object Detection". 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788.

Redmon, J., Farhadi, A. (2018). "YOLOv3: An Incremental Improvement". ArXiv, abs/1804.02767.

Rio-Alvarez, A., Suárez, J.D., Gonzalez-Rodriguez, M., Lanvin, D.F., Pérez, B.L. (2019). "Effects of Challenging Weather and Illumination on Learning-Based License Plate Detection in Noncontrolled Environments". Scientific Programming, 2019, 6897345:1-6897345:16.

Sajjad, K. M. (2010). "Automatic License Plate Recognition using Python and OpenCV". Pp:1-5

Sakib, S., Ahmed, N., Kabir, A.J., Ahmed, H. (2018). "An Overview of Convolutional Neural Network: Its Architecture and Applications".
<http://dx.doi.org/10.20944/preprints201811.0546.v1>

Smith, R.. (2007). "An Overview of the Tesseract OCR Engine". Ninth International Conference on Document Analysis and Recognition (ICDAR 2007). 2. 629 - 633. 10.1109/ICDAR.2007.4376991.

Srinath, K. R. (2017). "Python –The Fastest Growing Programming Language". In International Research Journal of Engineering and Technology (IRJET). Vol.4 Issue.12. pp: 354-357 <https://www.irjet.net/archives/V4/i12/IRJET-V4I1266.pdf>

Yaseen, A. F. (2018). "A Survey on the Layers of Convolutional Neural Networks". In International Journal of Computer Science and Mobile Computing, Vol.7 Issue.12, December- 2018, pg. 191-196

Zhang, Q. (2018). "Convolutional Neural Networks". In 3rd International Conference on Electromechanical Control Technology and Transportation - Volume 1: ICECTT, ISBN 978-989-758-312-4, pages 434-439. DOI: 10.5220/0006972204340439