

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Μάιος 2018

Ατομική Διπλωματική Εργασία

**ΑΠΟΚΑΤΑΣΤΑΣΗ ΜΕΡΙΚΩΣ ΕΠΙΚΑΛΥΜΜΕΝΩΝ ΕΙΚΟΝΩΝ
ΠΡΟΣΩΠΟΥ ΜΕ ΤΗΝ ΧΡΗΣΗ ΝΕΥΡΩΝΙΚΩΝ ΔΥΚΤΙΩΝ ΤΥΠΟΥ
ΠΕΡΙΟΡΙΣΜΕΝΩΝ ΜΗΧΑΝΩΝ BOLTZMANN**

Παναγιώτης Νικολάου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2018

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΠΟΚΑΤΑΣΤΑΣΗ ΜΕΡΙΚΩΣ ΕΠΙΚΑΛΥΜΜΕΝΩΝ ΕΙΚΟΝΩΝ ΠΡΟΣΩΠΟΥ ΜΕ ΤΗΝ ΧΡΗΣΗ ΝΕΥΡΩΝΙΚΩΝ ΔΥΚΤΙΩΝ ΤΥΠΟΥ ΠΕΡΙΟΡΙΣΜΕΝΩΝ ΜΗΧΑΝΩΝ BOLTZMANN

Παναγιώτης Νικολάου

Επιβλέπων Καθηγητής
Δρ. Χρίστος Χριστοδούλου

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

Μάιος 2018

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, Δρ. Χρίστο Χριστοδούλου, που μου έδωσε την ευκαιρία να εκπονήσω τη συγκεκριμένη διπλωματική. Με τη βοήθεια και τη σωστή καθοδήγηση που μου παρείχε όλο αυτό το διάστημα, καθώς και με την υπομονή που έχει υποδείξει, κατάφερα να φέρω εις πέρας μεγάλο κομμάτι της εργασίας. Επίσης, θα ήθελα να ευχαριστήσω τον κ. Αντρέα Λανίτη[14] για την βάση δεδομένων με τις εικόνες την οποία μας έδωσε για την εκπόνηση αυτής της εργασίας.

Περίληψη

Το θέμα της ατομικής διπλωματικής μου εργασίας είναι η εκπαίδευση ενός Deep Neural Network[1], με απώτερο σκοπό τον υπολογισμό της ολοκληρωμένης εικόνας προσώπου κάποιου ανθρώπου δοθείσας μιας εικόνας με αποκρυμμένη πληροφορία. Το πιο πάνω πρόβλημα επιστημονικά μπορούμε να το συναντήσουμε με τον όρο face occlusion[2]. Το πρόβλημα του face occlusion είναι ένα πολύ ενδιαφέρον θέμα που απασχολούσε από το παρελθόν τον τομέα της πληροφορικής. Αρκετές μελέτες έγιναν κατά διαστήματα σε αυτό τον τομέα κατά τις οποίες χρησιμοποιήθηκαν πολλά είδη νευρωνικών δικτύων για την επίλυση του προβλήματος αυτού.

Το δίκτυο που επιλέξαμε να χρησιμοποιήσουμε είναι τύπου Restricted Boltzmann Machine[3], η δομή του οποίου καθώς και περισσότερες λεπτομέρειες για αυτό αναλύονται σε βάθος στα επόμενα κεφάλαια. Με τον όρο occlusion εννοούμε την απόκρυψη πληροφοριών από κάποια δεδομένα. Στην δική μας περίπτωση τα δεδομένα είναι εικόνες προσώπων και οι πληροφορίες που είναι αποκρυμμένες είναι οι λεπτομέρειες του προσώπου όπως για παράδειγμα τα μάτια, η μύτη, το στόμα και το περίγραμμα του προσώπου. Σκοπός είναι να εκπαιδεύσουμε το δίκτυο RBM, με απώτερο σκοπό να μπορεί να συμπληρώνει την ελλείπουσα πληροφορία στα δεδομένα εισόδου. Το RBM που επιλέχθηκε για την λύση του πιο πάνω προβλήματος έχει την ιδιότητα της «κρυφής-συσχετιστικής μνήμης» την οποία στην περίπτωση του θέματος μας θα θέλαμε να εκμεταλλευτούμε. Συγκεκριμένα σκοπός μας είναι να εκπαιδεύσουμε το δίκτυο αυτό με μία τεράστια βάση δεδομένων από εικόνες προσώπων γεμίζοντας την «κρυφή-συσχετιστική μνήμη» με τις λεπτομέρειες από αυτές τις εικόνες. Έτσι στη συνέχεια δίνοντας στην είσοδο του δικτύου μια εικόνα προσώπου με αποκρυμμένες λεπτομέρειες το πλέον εκπαιδευμένο RBM θα βγάλει στην έξοδο το πιο όμοιο στοιχείο της μνήμης αυτής συμπληρώνοντας την εικόνα της εισόδου. Η εικόνα με τις ελλείψεις πληροφορίες προσώπου μπορεί να χαρακτηριστεί και ως εικόνα χαμηλών διαστάσεων. Αντιθέτως με το πιο πάνω, η επιθυμητή εικόνα εξόδου και η οποιαδήποτε εικόνα εκπαίδευσης

μπορεί να χαρακτηριστεί και ως εικόνα υψηλών διαστάσεων. Από το δίκτυο μας, επιθυμούμε να κατανοήσει την σχέση που κρύβεται πίσω από τις λεπτομέρειες των προσώπων των εικόνων εκπαίδευσης. Αν γίνει αυτό τότε το δίκτυο μας θα μπορεί να επιλύσει οποιαδήποτε εικόνα του δοθεί ως είσοδος.

Για να πετύχουμε τον σκοπό μας, δημιουργήσαμε δυο βάσεις δεδομένων από μαυρόασπρες εικόνες προσώπων και μια βάση δεδομένων από εικόνες με σημεία στα χαρακτηριστικά των προσώπων. Για κάθε μια από αυτές εκπαιδεύσαμε το δίκτυο με διάφορες παραμέτρους. Τα αποτελέσματα που λάβαμε από το δίκτυο ήταν σχετικά καλά στις βάσεις δεδομένων από τις εικόνες προσώπων και στην περίπτωση της βάσης δεδομένων των εικόνων από σημεία, τα αποτελέσματα ήταν πολύ καλά. Το μέσο τετραγωνικό σφάλμα μειώνεται αρκετά σε όλες τις περιπτώσεις. Από τα αποτελέσματα συμπεράναμε ότι το δίκτυο αποδίδει σε αυτό το θέμα ειδικά στις εικόνες από σημεία όπως φαίνεται και στις μαθηματικές μετρήσεις που πήραμε στις εικόνες από σημεία. Συγκεκριμένα το δίκτυο αποτυγχάνει μόνο σε ελάχιστα εικονοστοιχεία σε κάθε περίπτωση, κάτι που είναι αρκετά ενθαρρυντικό. Από τις δύο πρώτες βάσεις που δημιουργήσαμε παρατηρήσαμε ότι το δίκτυο αντιλαμβάνεται και συμπληρώνει το occlusion στις εικόνες και τις συμπληρώνει στις πλείστες περιπτώσεις, όμως σε αρκετά από τα αποτελέσματα μας είχαμε αρκετό μη επιθυμητό θόρυβο, Γι' αυτόν το λόγο δοκιμάσαμε το δίκτυο μας στις εικόνες προσώπων με σημεία. Τα αποτελέσματα ήταν αρκετά ενθαρρυντικά αν αναλογιστούμε ότι χρησιμοποιήσαμε όσο το δυνατό μικρότερες εικόνες μπορούσαμε λόγω πίεσης χρόνου και έλλειψή υπολογιστικής ισχύος. Η συμπίεση των εικόνων σε μικρότερες διαστάσεις ήταν ο λόγος του θορύβου στα αποτελέσματα με δεδομένα τις πρώτες δύο βάσεις δεδομένων. Αξίζει να σημειωθεί ότι οι μαυρόασπρες εικόνες που χρησιμοποιήσαμε ήταν από την συλλογή του κ. Λανίτη[14] τροποποιημένες σε μαυρόασπρες με δύο διαφορετικές τιμές κατωφλίου για την κάθε βάση που φτιάξαμε.

Περιεχόμενα

Κεφάλαιο 1	1
Εισαγωγή	1
1.1 Ορισμός και σημασία του προβλήματος	1
1.2 Περιγραφή προηγούμενης εργασίας	3
Κεφάλαιο 2	11
Δεδομένα εισόδου	11
2.1 Σύντομη περιγραφή δεδομένων εισόδου	11
Κεφάλαιο 3	14
Γνωσιολογικό υπόβαθρο	14
3.1.1 Γενικά	14
3.1.2 Δομή	15
3.1.3 Οι νευρώνες των δύο επιπέδων	15
3.1.4 Βάρη	15
3.1.5 Έξοδος Νευρώνων	16
3.1.6 Αλγόριθμος Εκπαίδευσης	17
3.1.7 Το δίκτυο από Μαθηματική/Φυσική σκοπιά	18
Σχεδιασμός και Υλοποίηση Νευρωνικών Δικτύων	21
4.1 Προ επεξεργασία δεδομένων ελέγχου	21
4.2 Στατιστικές μέθοδοι για υπολογισμό σφάλματος	26
Κεφάλαιο 5	29
Αποτελέσματα και Συζήτηση	29
5.1 Τρόπος Αξιολόγησης αποτελεσμάτων	29
5.2 Αποτελέσματα MNIST Unlabeled Handwritten Digits	30
5.3 Αποτελέσματα training set 1	31
Πείραμα 1 ^ο	32
Πείραμα 2 ^ο	35
Πείραμα 3 ^ο	38
Πείραμα 4 ^ο	40
Πείραμα 5 ^ο	42
5.4 Αποτελέσματα training set 2	45
Πείραμα 1 ^ο	45
Πείραμα 2 ^ο	47
Πείραμα 3 ^ο	50
Πείραμα 4 ^ο	52
5.5 Αποτελέσματα με τις εικόνες σημείων	54

Πείραμα 1 ^ο :	54
Πείραμα 2 ^ο :	55
Πείραμα 3 ^ο :	55
Κεφάλαιο 6.....	57
Συμπεράσματα	57
Κεφάλαιο 7.....	58
Μελλοντική Εργασία	58
ΑΝΑΦΟΡΕΣ.....	60
Παράρτημα – Κώδικας	62

Κεφάλαιο 1

Εισαγωγή

1.1 Ορισμός και σημασία του προβλήματος

1.2 Περιγραφή προηγούμενης εργασίας

1.1 Ορισμός και σημασία του προβλήματος

Αρχικά, με τον όρο face occlusion αναφερόμαστε στο πρόβλημα κατά το οποίο πληροφορίες από μια εικόνα προσώπου απουσιάζουν. Στον τομέα της πληροφορικής, αυτό που μας ενδιαφέρει είναι με την χρήση της τεχνολογίας και διάφορων αλγορίθμων να μπορέσουμε να συμπληρώσουμε την οποιαδήποτε ελλείπουσα πληροφορία από την εικόνα προσώπου που θα δοθεί στην είσοδο του προγράμματος. Όταν αναφερόμαστε σε πληροφορία εικόνας προσώπου εννοούμε τις λεπτομέρειες που εμπεριέχονται σε ένα ανθρώπινο πρόσωπο. Οι λεπτομέρειες αυτές μπορεί να είναι τα μάτια, τα φρύδια, το στόμα, η μύτη καθώς και το περίγραμμα του προσώπου. Εάν ένα σύνολο από τα πιο πάνω απουσιάζει τότε η εικόνα του προσώπου θεωρείται ελλιπής – occluded. Αυτού του είδους τις εικόνες προσπαθούμε να συμπληρώσουμε στο θέμα μας. Η επιθυμητή εικόνα εξόδου είναι μια εικόνα πανομοιότυπη με αυτή της εισόδου με συμπληρωμένα τα ελλείπων στοιχεία.

Το face occlusion είναι ένα πρόβλημα που απασχολούσε το τομέα της πληροφορικής για πολλά χρόνια. Μπορεί να φανεί χρήσιμο σε περιπτώσεις εγκλήματος κατά τις οποίες αναζητάτε ο εγκληματίας, το πρόσωπο του οποίου ήταν μερικώς επικαλυμμένο από μια εικόνα που λήφθηκε στο περιστατικό. Πέρα

από την εγκληματολογική σκοπιά το face occlusion μπορεί να φανεί αρκετά βοηθητικό σε εφαρμογές face detection, όπου το πρόσωπο έχει κλίση από τον φακό ή είναι μερικός επικαλυμμένο. Αυτού του είδους οι εφαρμογές χρησιμοποιούνται σε εφαρμογές ασφαλείας που εφαρμόζουν face detection για να δώσουν πρόσβαση σε ένα συγκεκριμένο πρόσωπο.

Επιπρόσθετα, στην κατηγορία του face occlusion ανήκει και το πρόβλημα του face progression που μπορεί να το συναντήσουμε και με την ονομασία face aging. Το face aging είναι το πρόβλημα κατά το οποίο σκοπός είναι η πρόβλεψη της μελλοντικής εικόνας του προσώπου που δίνεται ως είσοδος. Η μελλοντική εικόνα ονομάζεται και εικόνα γερασμένου προσώπου. Η εικόνα του νεαρού προσώπου μπορεί να χαρακτηριστεί και ως εικόνα χαμηλών διαστάσεων και αντιθέτως με το πιο πάνω, η εικόνα του γερασμένου προσώπου μπορεί να χαρακτηριστεί και εικόνα υψηλών διαστάσεων. Η λογική πίσω από τα πιο πάνω είναι ότι η εικόνα του γερασμένου προσώπου περιέχει λεπτομέρεια της νεαρής εικόνας προσώπου που είναι αποκρυμμένη από την εικόνα του νεαρού προσώπου. Το face aging είναι και αυτό ένα πολύ σημαντικό πρόβλημα αφού μπορεί να φανεί αρκετά χρήσιμο και να βοηθήσει τους ερευνητές να λύσουν εγκλήματα και να βρουν αγνοούμενα άτομα (περιπτώσεις εξαφανισμένου προσώπου που απουσιάζει για μεγάλο χρονικό διάστημα).

Τα Restricted Boltzmann machines μπορούν να χρησιμοποιηθούν στην αναγνώριση ελλιπών προσώπων (occluded), αφού με την χρήση της «κρυφής-συσχετιστικής μνήμης» που διαθέτουν μπορούν να βγάλουν στην έξοδο, τα χαρακτηριστικά που απουσιάζουν από την εικόνα εισόδου, με βάση τα υπάρχοντα χαρακτηριστικά στην εικόνα εισόδου, και τα χαρακτηριστικά των δεδομένων εκπαίδευσης που το δίκτυο απομνημόνευσε κατά την φάση εκπαίδευσης. Το πρόβλημα του face occlusion και του age progression είναι ένα θέμα που απασχόλησε αρκετές φορές στο παρελθόν τον χώρο της πληροφορικής και όχι μόνο. Προτάθηκαν και εφαρμόστηκαν αρκετές τεχνικές οι οποίες περιγράφονται αναλυτικά στην επόμενη παράγραφο.

1.2 Περιγραφή προηγούμενης εργασίας

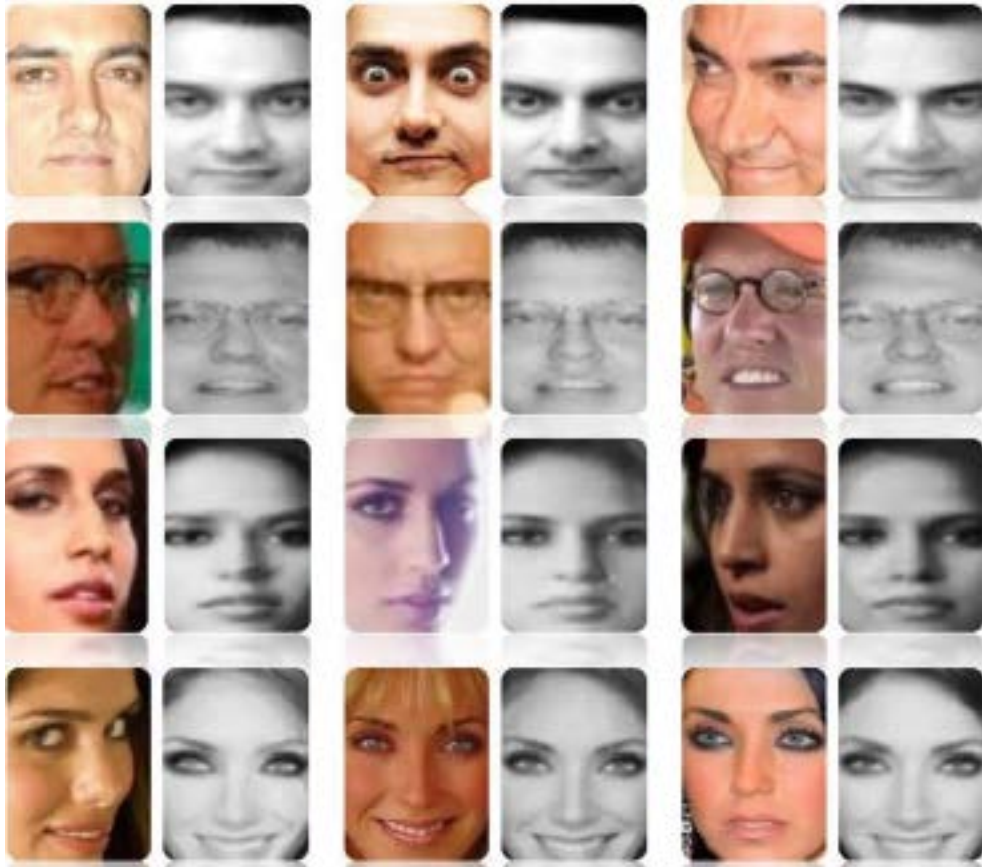
Κατά την φάση της αρχικής μου έρευνας στο θέμα, εντόπισα αρκετές προηγούμενες εργασίες τόσο στον τομέα του face occlusion όσο και στο θέμα του age progression. Το τελικό μας θέμα αφορούσε face occlusion, όμως αξίζει να αναφερθούμε και σε μερικές εργασίες στον τομέα του age progression.

Αρχικά, όσο αφορά το face occlusion εντόπισα την εργασία των Yi Sun et al. Συγκεκριμένα, σε άρθρο τους το 2013 με τίτλο Deep Convolutional Network Cascade for Facial Point Detection[4], πρότειναν μια νέα μέθοδο για τον υπολογισμό των θέσεων των βασικών σημείων του προσώπου με την χρήση convolutional networks τριών επιπέδων. Τα βασικά σημεία οι ίδιοι τα αποκαλούσαν σημεία κλειδιά. Σε κάθε επίπεδο, συγχωνεύονταν έξοδοι από πολλαπλά δίκτυα για πιο ακριβή εκτίμηση των σημείων κλειδιών. Λόγω της υψηλής ακρίβειας των convolutional networks, αρκετά χαρακτηριστικά του προσώπου εξάγονται σε ολόκληρη την περιοχή του προσώπου κατά το στάδιο της αρχικοποίησης. Αυτό έχει σαν πλεονέκτημα το ότι αυτές οι πληροφορίες θα χρησιμοποιηθούν για να τοποθετηθεί το κάθε σημείο κλειδί. Επιπλέον, ένα ακόμη πλεονέκτημα είναι ότι δεδομένου ότι τα δίκτυα εκπαιδεύονται για να προβλέπουν ταυτόχρονα όλα τα σημεία-κλειδιά, οι γεωμετρικοί περιορισμοί μεταξύ των σημείων-κλειδιών κωδικοποιούνται σιωπηρά. Με το πιο πάνω εννοούμε ότι δεν είναι όλα τα πρόσωπα ίδια. Και αυτό συνεπάγεται ότι δεν μπορεί το δίκτυο απλά να απομνημονεύσει τα σημεία σε κάθε πρόσωπο. Αυτό που πραγματικά συμβαίνει είναι το ότι το δίκτυο βρίσκει τρόπο να προβλέπει την θέση του σημείου που λείπει σε συνδυασμό με τις θέσεις των υπόλοιπων σημείων. Επομένως, η μέθοδος μπορεί να αποφύγει τοπικό ελάχιστο που προκαλείται από αμφισημία και καταστροφή δεδομένων σε δύσκολα δείγματα εικόνας λόγω απόφραξης(occlusion), μεγάλων παραλλαγών θέσεων και ακραίων φωτισμών. Τα δίκτυα στα επόμενα δύο επίπεδα εκπαιδεύονται για να βελτιώσουν τοπικά τις αρχικές προβλέψεις και οι εισροές τους περιορίζονται σε μικρές περιοχές γύρω από τις αρχικές προβλέψεις.



Εικόνα 1.2.1 [4] Παραδείγματα ανίχνευσης σημείων προσώπου όπως παρουσιάζετε στο άρθρο τους. Βλέπουμε σε μερικές εικόνες ότι αν και οι εικόνες περιέχουν αρκετή απόκρυψη πληροφορίας, το δίκτυο τους εντοπίζει τα σωστά σημεία κλειδιά.

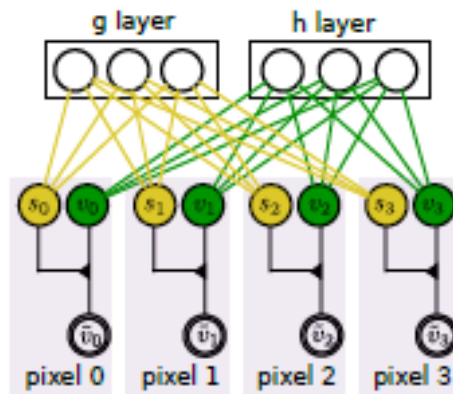
Η επόμενη εργασία που εντόπισα στο face occlusion είναι αυτή των Zhenyao Zhu et al οι οποίοι πρότειναν μια μέθοδο για την αντιμετώπιση των διάφορων παρεμβολών που δημιουργούνται κατά την λήψη μιας εικόνας προσώπων. Περισσότερες λεπτομέρειες για την μέθοδο τους περιγράφουν στο άρθρο τους με τίτλο Recover Canonical-View Faces in the Tild with Deep Neural Networks [5]. Και αυτοί με την σειρά τους έκαναν χρήση convolutional networks. Απώτερος σκοπός της εργασίας τους ήταν να συμπληρώνει οποιαδήποτε λεπτομέρεια απουσίαζε. Σημαντικό κομμάτι στην εργασίας τους είναι ότι η τελική εικόνα που παράγε το δίκτυο τους βρισκόταν σε μορφή portrait.



Εικόνα 2.2.2 [5] Η προτεινόμενη μέθοδος μπορούσε να ανακτήσει από εικόνες κανονικής προβολής και φωτισμού μέχρι και από εικόνες με μεγάλη απουσία πληροφοριών. Για παράδειγμα, σε μια σειρά, εμφανίζονται οι εικόνες εισόδου και δίπλα οι εικόνες ανακατασκευής τους.

Μία ακόμη ενδιαφέρουσα δουλειά στον τομέα του face occlusion είναι αυτή που παρουσιάζεται από τους Yichuan Tang et al το 2012 σε άρθρο τους με τίτλο Robust Boltzmann machines for Recognition and Denoising[6]. Στην εργασία τους χρησιμοποιούν μια παραλλαγμένη μορφή των Boltzmann machines που ονομάζετε Robust Boltzmann machine. Όπως οι ίδιοι ισχυρίζονται, αυτή η μορφή δικτύου επιτρέπει στα Boltzmann machines να είναι πιο ανθεκτικά σε κατεστραμμένα - occluded δεδομένα και στον θόρυβο. Ακολουθεί περιγραφή του novel μοντέλου που πρότειναν. Όπως περιγράφουν στον άρθρο τους, το δίκτυο είναι ένας μη κατευθυνόμενο γραφικό μοντέλο που αποτελείται από 3 συστατικά. Το πρώτο είναι ένα Gaussian Restricted Boltzmann machine και είναι υπεύθυνο για την μοντελοποίηση της πυκνότητας

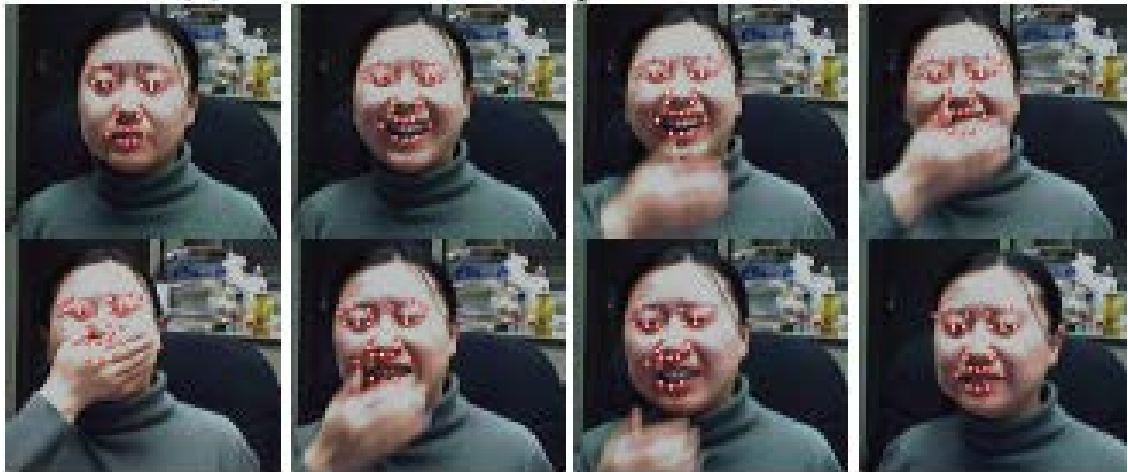
των θορύβων στα δεδομένα. Το δεύτερο είναι ένα απλό Restricted Boltzmann machine το οποίο είναι υπεύθυνο για την μοντελοποίηση της μορφής των αποκρυμμένων πληροφοριών στα δεδομένα. Το τρίτο συστατικό του μοντέλου τους είναι ένας μηχανισμός ο οποίος είναι αυτός που επιτρέπει στο μοντέλο να είναι ανθεκτικό στις παρεμβολές και τους θορύβους. Την γραφική αναπαράσταση του μοντέλου τους μπορούμε να την δούμε στην εικόνα 1.2.3



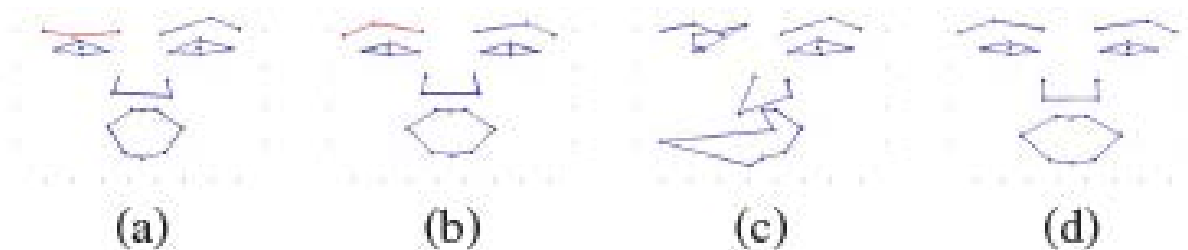
Εικόνα 3.2.3 [6] Γραφική απεικόνιση του μοντέλου τους, όπως οι ίδιοι το παρουσίασαν στο άρθρο τους. Τα σκιασμένα τρίγωνα υποδεικνύουν ότι το s_i λαμβάνει την τιμή από την ένωση του v_i με το $\sim v_i$. Οι κίτρινες ενώσεις είναι τα βάρη του RBM ενώ τα πράσινα είναι οι ενώσεις με το GRBM. Το κάθε εικονοστοιχείο μοντελοποιείται από τρεις τυχαίες μεταβλητές v_i , $\sim v_i$ και s_i .

Τελευταία εργασία που εντόπισα στον τομέα του face occlusion είναι αυτή των Yue Wu et al το 2013, η εργασία των οποίων παρουσιάζεται σε άρθρο τους με τίτλο Facial Feature Tracking under Varying Facial Expressions and Face Poses based on Restricted Boltzmann Machines[7]. Χρησιμοποίησαν ένα 3way Restricted Boltzmann machine το οποίο εκπαίδευσαν με απώτερο σκοπό να συλλαμβάνει την σχέση μεταξύ μετωπικών σχημάτων προσώπων και μη μετωπικών σχημάτων προσώπων. Με τις πιο πάνω εκφράσεις εννοούμε εικόνες που τραβήχτηκαν απέναντι από τον φακό και εικόνες που είχαν κλίση από τον φακό και δεν ήταν εμφανές όλη η επιφάνεια του προσώπου στην εικόνα. Τέλος, εισάγουν κάποιες μεθόδους για να συνδυάζουν συμμετρικά τα βασικά σχήματα μοντέλων προσώπου με κάποιες μετρήσεις στις εικόνες για τα βασικά σημεία των χαρακτηριστικών των προσώπων. Συγκεκριμένα σκοπός τους ήταν να εντοπίζουν 26 σημεία στο ανθρώπινο πρόσωπο. Στις εικόνες που

ακολουθούν βλέπουμε τα 26 σημεία που εντόπιζαν στο πρόσωπο, καθώς και τις ελλιπής εικόνες που συμπληρώνεται.



Εικόνα 4.2.4 [7] Βλέπουμε τα 26 σημεία στο πρόσωπο τα οποία προσπαθούσαν να εντοπίσουν.



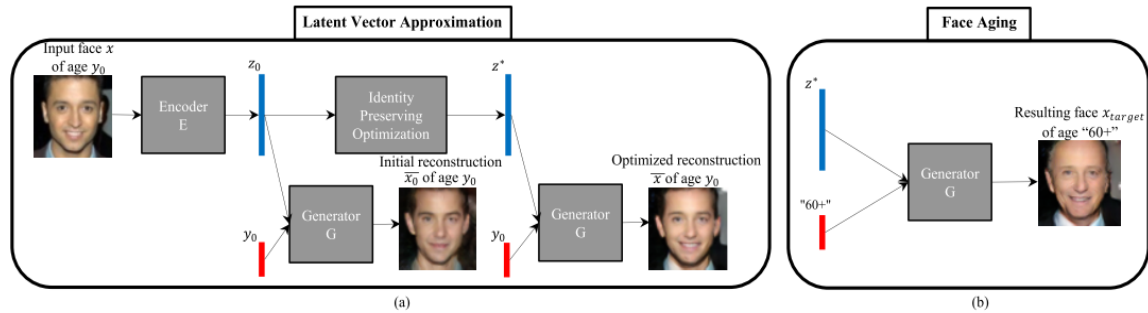
Εικόνα 5.2.5 [7] Βλέπουμε στο (a) μια εικόνα με στραβωμένο φρύδι, το οποίο διορθώνεται στο (b). Ανάλογα βλέπουμε μία εικόνα με occlusion στο (c) το οποίο διορθώνεται στο (d).

Από την άλλη αξίζει να αναφερθούμε και σε μερικές προηγούμενες εργασίες στον τομέα του age progression.

Αρχικά, ένα άρθρο από τους Xiangbo Shu et al[8], περιγράφει την δική τους εργασία σε αυτό τον τομέα. Γενικά, δημιούργησαν μια μέθοδο κατά την οποία παίρνοντας ως είσοδο μια εικόνα προσώπου, βγάζει στην έξοδο του δικτύου την εικόνα του προσώπου που θα ήταν μετά απο την πάροδο μίας χρονικής περιόδου. Το πώς το πέτυχαν αυτό περιγράφεται πιο κάτω. Αρχικά,

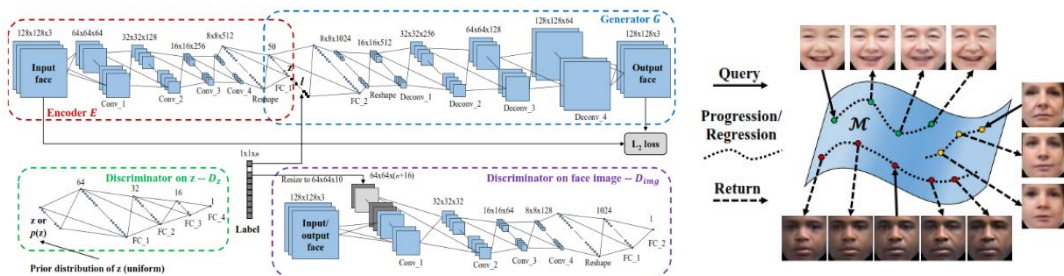
έσπασαν τις ηλικίες σε ομάδες σταθερού ηλικιακού διαστήματος (π.χ. 0-5, 5-10 κ.λπ.) και η κάθε ομάδα είχε το δικό της λεξικό που την χαρακτήριζε. Τα λεξικά αυτά εκπαιδεύτηκαν με βάση κάποιες εικόνες προσώπων. Έτσι μετά από την εκπαίδευση, η κάθε ηλικιακή ομάδα είχε ένα σύνολο χαρακτηριστικών που την χαρακτήριζε. Η πιο πάνω διαδικασία ήταν το «offline phase», όπως την ονόμασαν, και αφορούσε τα σταθερά χαρακτηριστικά γήρανσης-λεξικά γήρανσης της κάθε ηλικιακής ομάδας. Η επόμενη φάση, ήταν το «online phase» και αφορούσε τα εξατομικευμένα χαρακτηριστικά του κάθε προσώπου δηλαδή σημάδια στο πρόσωπο, ελιές, κ.λπ. Η εκπαίδευση του συστήματος έγινε με εικόνες από δύο βάσεις δεδομένων, την CASD[9] και την MORPH[10]. Σε αυτές τις δύο βάσεις υπήρχαν εικόνες προσώπων διάφορων ατόμων για χρονικό διάστημα μικρότερο των 20 χρόνων. Έτσι το λεξικό κάθε ηλικιακής ομάδας εκπαιδεύτηκε με τις ανάλογες εικόνες (short term face aging sequences). Για την συλλογή των εικόνων προσώπου χρησιμοποιήθηκε η μέθοδος του face alignment. Η δοκιμή του συστήματος έγινε με εικόνες από την βάση δεδομένων FGNET[11]. Τα πιο πάνω αφορούσαν την εκπαίδευση του δικτύου. Μετά από την εκπαίδευση έδιναν μια εικόνα στο σύστημα και έβγαζε την εικόνα που θα είχε στην επόμενη ηλικιακή ομάδα. Έτσι επαναλάμβαναν την ίδια διαδικασία με την προηγούμενη εικόνα εξόδου του συστήματος ως την νέα είσοδο σε αυτό, μέχρι να παραχθούν οι εικόνες για την κάθε ηλικιακή ομάδα.

Μια άλλη σχετική εργασία στο Age procession περιγράφεται σε ένα άρθρο γραμμένο από τους Grigory Antipov et al το 2017[12], όπου περιγράφεται η εργασία τους στο face aging. Οι πιο πάνω, έφτιαξαν μια μέθοδο η οποία παίρνοντας ως είσοδο μια εικόνα προσώπου, βγάζει στην έξοδο την εικόνα του προσώπου που θα ήταν μετά το πέρας μιας χρονικής περιόδου (ytarget). Για την λειτουργία του δικτύου τους χρησιμοποίησαν ένα Conditional Generative Adversarial Network(GAN), όπου με βάση κάποια optimizations επεξεργαζόταν κάποια vectors και εν τέλει παραγόταν η τελική εικόνα. Περισσότερες λεπτομέρειες για τον τρόπο λειτουργίας περιγράφονται στην εικόνα 1.2.6



Εικόνα 6.2.6 [12] Βλέπουμε την γραφική αναπαράσταση της δουλειάς των Grigory Antipov et al το 2017. μπορούμε να δούμε την γραφική αναπαράσταση του τρόπου λειτουργίας του δικτύου τους. Όσο αφορά το (a), χρησιμοποίησαν Conditional Generative Adversarial Networks (GANs). Εκπαίδευσαν ένα Encoder E , ο οποίος παίρνοντας ως είσοδο την φωτογραφία εισόδου, παράγει ένα Latent vector z_0 . Το E εκπαιδεύτηκε έτσι ώστε να ελαχιστοποιεί την ευκλείδεια απόσταση μεταξύ του υπολογισμένου latent vector και του πραγματικού latent vector. Στην συνέχεια το z_0 επεξεργαζόταν από κάποια optimizations και έτσι παραγόταν το z^* latent vector. Έτσι, με την χρήση του z_0 , δημιουργείται η αρχική εικόνα $\bar{x}_0$, και με το z^* δημιουργείται η βελτιστοποιημένη εικόνα. Έπειτα από αυτά, για το (b), με είσοδο το z^* και την επιθυμητή ηλικία, παράγεται η τελική εικόνα.

Τέλος, μια ακόμη δουλειά που αξίζει να αναφερθεί είναι αυτή που περιγράφεται από τους Zhifei Zhang et al στο άρθρο τους το 2017 [13]. Συγκεκριμένα, προτάθηκε ένα Conditional Adversarial Autoencoder (CAAE), το οποίο μαθαίνει μια πολλαπλή προσπέλαση προσώπου (face manifold), που διασχίζει την ομαλή εξέλιξη της ηλικίας και την παλινδρόμηση που μπορεί να πραγματοποιηθεί ταυτόχρονα. Στο CAAE, το πρόσωπο αρχικά χαρτογραφείται σε ένα latent vector z , μέσω ενός κωδικοποιητή, και στη συνέχεια το vector προβάλλεται στο face manifold αναλόγως της ηλικίας, μέσω ενός deconvolutional generator. Η πιο πάνω διαδικασία γίνεται διότι είναι δύσκολο να χειριστεί κανείς το face manifold. Το latent vector, διατηρεί τα εξατομικευμένα χαρακτηριστικά προσώπου (δηλ. Προσωπικότητα). Δύο αντίπαλα δίκτυα (GANs), όπως μπορούμε να διακρίνουμε και από την εικόνα 1.2.7, χρησιμοποιούνται στον encoder και στον generator, αντιστοίχως, βοηθώντας το σύστημα να παράγει περισσότερα φωτορεαλιστικά πρόσωπα. Τα δεδομένα για εκπαίδευση συλλέχθηκαν από τις βάσεις δεδομένων MORPH [9] και CACD [10].



Εικόνα 7.2.7 [13] Στο κόκκινο ορθογώνιο φαίνεται η λειτουργία του encoder. Δίπλα, στο γαλάζιο ορθογώνιο, βλέπουμε την λειτουργία του Generator G, όπου παίρνει το latent vector z του Encoder, και σε συνδυασμό με την ηλικία βγάζει την τελική εικόνα προσώπου. Κάτω, στο πράσινο ορθογώνιο βλέπουμε την λειτουργία του discriminator D_z , όπου παίρνει ως είσοδο το z , και εφαρμόζει ομοιόμορφη κατανομή σε αυτό. Τέλος στο μωβ ορθογώνιο βλέπουμε το D_{img} , όπου κάνει την έξοδο να είναι πιο φωτορεαλιστική. Τα D_{img} και D_z , είναι τα δίκτυα που ανταγωνίζονται άρα κληρονομούνται οι ιδιότητες των Generative Adversarial Networks (GANs).

Κεφάλαιο 2

Δεδομένα εισόδου

2.1 Σύντομη περιγραφή δεδομένων εισόδου

2.1 Σύντομη περιγραφή δεδομένων εισόδου

Τα δίκτυα RBMs χρησιμοποιήθηκαν αρκετές φορές στο παρελθόν για συμπλήρωση ελλείπουσας πληροφορίας σε δεδομένα. Διάφορες μορφές δεδομένων τα οποία χρησιμοποιήθηκαν στα δίκτυα RBMs είναι οι εικόνες και οι ήχοι. Αρχικά, προσπαθήσαμε να δούμε την έννοια της ελλείπουσας πληροφορίας (occluded) από μια άλλη οπτική γωνία. Συγκεκριμένα, η ιδέα μας ήταν να δώσουμε δεδομένα με επιπλέον πληροφορία στο δίκτυο κατά την εκπαίδευση του, την οποία θα περιμέναμε να συμπληρώσει κατά την φάση δοκιμής σε δεδομένα με ελλείπουσα πληροφορία (occluded data). Έτσι αρχική μας ιδέα ήταν να ασχοληθούμε με το θέμα του Age progression. Συγκεκριμένα, θα βάζαμε εικόνες γερασμένων προσώπων στα δεδομένα εκπαίδευσης και εικόνες νεαρών προσώπων στα δεδομένα δοκιμής. Με άλλα λόγια, στα δεδομένα δοκιμής θα απουσίαζαν οι πληροφορίες που χαρακτηρίζουν την γήρανση του ανθρώπινου προσώπου. Η αναμενόμενη έξοδος του δικτύου στην φάση δοκιμής θα ήταν η συμπλήρωση της ελλείπουσας πληροφορίας γήρανσης στην εικόνα του νεαρού προσώπου που θα δινόταν ως είσοδος. Βασική προϋπόθεση όμως για να επιτύχει το πιο πάνω είναι ο τεράστιος όγκος δεδομένων σε εικόνες, οι οποίες να ήταν από γερασμένα πρόσωπα, έτσι ώστε μετά την εκπαίδευση του δικτύου να απομνημονεύσει τις λεπτομέρειες που υπάρχουν στα γερασμένα πρόσωπα όπως για παράδειγμα οι ρυτίδες. Στην

συνέχεια όμως διαπιστώσαμε ότι οι νευρώνες των Restricted Boltzmann machines είναι δυαδικοί. Έτσι δεν μπορούσαμε να εργαστούμε στο θέμα του face progression. Εν τέλει αποφασίσαμε να εργαστούμε στο τομέα του face occlusion με δυαδικές εικόνες.

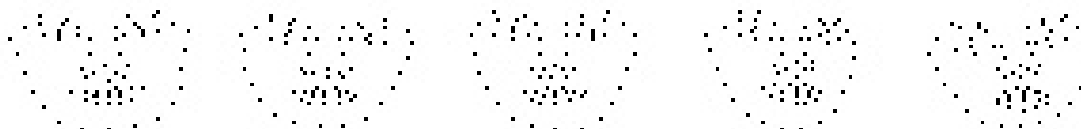
Έτσι, θα έπρεπε να δημιουργήσουμε ή να βρούμε μία τεράστια βάση δεδομένων από εικόνες προσώπων. Αυτές τις εικόνες θα έπρεπε να τις επεξεργαστούμε ούτως ώστε να τις φέρουμε στην επιθυμητή μορφή που να μπορεί πρώτα να εκπαιδευτεί επαρκώς το δίκτυο χωρίς επιπλέον παρεμβολές και κατά δεύτερο να μπορούν να κωδικοποιηθούν και να αποκωδικοποιηθούν εύκολα πριν και μετά την εκπαίδευση. Τα δεδομένα μας θα έπρεπε να ανήκουν στην κατηγορία των unlabeled δεδομένων αφού δεν μπορούν να χαρακτηριστούν από κάποια ετικέτα. Τα δεδομένα που χρησιμοποιήσαμε ήταν η βάση δεδομένων του κ. Λανίτη[14], όπου ήταν εικόνες σε greyscale μορφή.

Οι εικόνες που λάβαμε από τον κ. Λανίτη[14] ήταν ήδη προεπεξεργασμένες σε μεγάλο βαθμό, έτσι έγινε πιο εύκολο και το δικό μας έργο στην συνέχεια. Σε αυτή την παράγραφο θα περιγράψουμε την προεπεξεργασία που υπήρχε ήδη στις εικόνες. Για την επεξεργασία που κάναμε εμείς θα μιλήσουμε στο κεφάλαιο για την προεπεξεργασία των δεδομένων. Συγκεκριμένα οι εικόνες του κ. Λανίτη[14] ήταν αποκομμένες γύρω από το περίγραμμα του προσώπου δηλαδή οποιασδήποτε λεπτομέρεια δεν αφορούσε το πρόσωπο αποκόπηκε και αυτό μας βοήθησε αρκετά. Όταν αναφερόμαστε σε λεπτομέρειες εκτός προσώπου εννοούμε τα αυτιά, τα μαλλιά και το παρασκήνιο της εικόνας. Σημαντικό ήταν το γεγονός ότι μέσα στην βάση δεδομένων υπήρχαν εικόνες από όλες τις ηλικίες και από τα δύο φύλα. Αυτό είναι αρκετά σημαντικό γιατί θα είχαμε μια βάση από εικόνες με πλούσιες εικόνες από όλες τις κατηγορίες έτσι το δίκτυο μας θα είχε την δυνατότητα να εκπαιδευτεί αρκετά καλά. Στην εικόνα 2.1.1 μπορούμε να κατανοήσουμε και γραφικά για την προεπεξεργασία που υπήρχε ήδη στις εικόνες που λάβαμε από τον κ. Λανίτη[14].



Εικόνα 2.1.1 [14] Πιο πάνω βλέπουμε τις φωτογραφίες από την βάση δεδομένων. Βλέπουμε ότι βρίσκονται σε μορφή Greyscale και ότι το μόνο που κρατήθηκε από την εικόνα είναι μόνο το πρόσωπο.

Εκτός από τις δύο βάσεις δεδομένων δοκιμάσαμε και το δίκτυο μας στην αντίστοιχη βάση δεδομένων των πιο πάνω εικόνων αλλά με σημεία στα πιο σημαντικά σημεία του προσώπου. Στην εικόνα 2.1.2 μπορούμε να δούμε αυτό που περιγράφουμε πιο πάνω. Επίσης στην εικόνα 2.1.3 μπορούμε να δούμε τα 68 σημεία σε μια κανονική εικόνα ανθρώπινου προσώπου.



Εικόνα 2.1.2 Πιο πάνω βλέπουμε τις φωτογραφίες από την βάση δεδομένων που δημιουργήσαμε από τα αρχεία του κ. Λανίτη που περιείχαν συντεταγμένες των βασικών σημείων του κάθε προσώπου. Συγκεκριμένα η κάθε εικόνα περιείχε 68 σημεία.



Εικόνα 2.1.3 Πιο πάνω βλέπουμε τα 68 βασικά σημεία να αναπαρίστανται πάνω σε μια κανονική ανθρώπινη φωτογραφία προσώπου.

Κεφάλαιο 3

Γνωσιολογικό υπόβαθρο

3.1 Δίκτυα Restricted Boltzmann Machines

- 3.1.1 Γενικά
- 3.1.2 Δομή
- 3.1.3 Οι νευρώνες των δύο επιπέδων
- 3.1.4 Βάρη
- 3.1.5 Έξοδος νευρώνων
- 3.1.6 Αλγόριθμος εκπαίδευσης

3.1.1 Γενικά

Τα RBMs είναι νευρωνικά δίκτυα τα οποία ανέκαθεν χρησιμοποιούνταν ως generative models για διάφορων ειδών δεδομένα τόσο labelled όσο και unlabeled. Τα δίκτυα RBMs είναι πανομοιότυπα με τα Boltzmann machines[17] με την διαφορά ότι στα RBMs εξαλείφονται οι συνδέσεις μεταξύ νευρώνων που ανήκουν στο ίδιο layer. Με αυτό τον περιορισμό - restriction η εκπαίδευση μπορεί να γίνει παράλληλα και επομένως πιο γρήγορα. Τα απλά Boltzmann machines ήταν πολύ αργά στην εκπαίδευση, γι' αυτό μετά από την ανακάλυψη των Restricted Boltzmann machines άρχισαν να γίνονται πολύ γρήγορα δημοφιλή και άρχισαν να χρησιμοποιούνται ευρέως. Επιπρόσθετα, κατά διαστήματα προτάθηκαν διαφορετικοί αλγόριθμοι για την εκπαίδευση του δικτύου. Περισσότερες πληροφορίες ακολουθούν στις επόμενες παραγράφους.

3.1.2 Δομή

Όσο αφορά την δομή του, ένα δίκτυο RBM αποτελείται από δυο layers. Το πρώτο layer ονομάζεται visible layer και η λειτουργία του είναι να εισάγει τα δεδομένα μέσα στο δίκτυο. Επιπρόσθετα, χρησιμοποιείται τόσο και στην φάση εκπαίδευσης αλλά και στην φάση δοκιμής. Περισσότερες πληροφορίες για αυτό στην επόμενη παράγραφο. Το δεύτερο layer ονομάζεται hidden layer, και είναι αυτό που λειτουργεί σαν ανιχνευτής χαρακτηριστικών. Οι διαφορές των δύο layers είναι η λειτουργικότητα των νευρώνων τους. Περισσότερα και γι' αυτό στην επόμενη παράγραφο. Τα δύο layers μεταξύ τους συνδέονται με βάρη τα οποία μπορούν να χαρακτηριστούν ως μη κατευθυνόμενα ή και bidirectional.

3.1.3 Οι νευρώνες των δύο επιπέδων

Όπως αναφέρθηκε και πιο πάνω, το Hidden Layer είναι αυτό που είναι υπεύθυνο να εισάγει τα δεδομένα μέσα στο δίκτυο. Οι νευρώνες αυτού του Layer χρησιμοποιούνται τόσο στην φάση δοκιμής όσο και στην φάση εκπαίδευσης. Στην φάση εκπαίδευσης, σύμφωνα με τον αλγόριθμο Contrastive Divergence, που χρησιμοποιήσαμε για την εκπαίδευση του δικτύου μας, οι εξόδοι του κρυφού επιπέδου κατά την negative φάση του αλγορίθμου γίνονται οι εισόδοι μέσα στο δίκτυο με σκοπό να υπολογισθεί το reconstruction error της κάθε εποχής. Για να γίνει αυτό, τα βάρη του δικτύου αλλάζουν φορά και δεδομένα εισόδου είναι οι εξόδοι των νευρώνων του Hidden Layer. Ουσιαστικά τα δύο layers αλλάζουν ρόλους και αντί για δεδομένα εισόδου από τα δεδομένα εκπαίδευσης, δίνεται στο δίκτυο το output του δικτύου για το συγκεκριμένο δεδομένο εισόδου. Έτσι, μπορούμε να εξάγουμε το συμπέρασμα ότι τα δύο layers έχουν πανομοιότυπο τρόπο λειτουργίας.

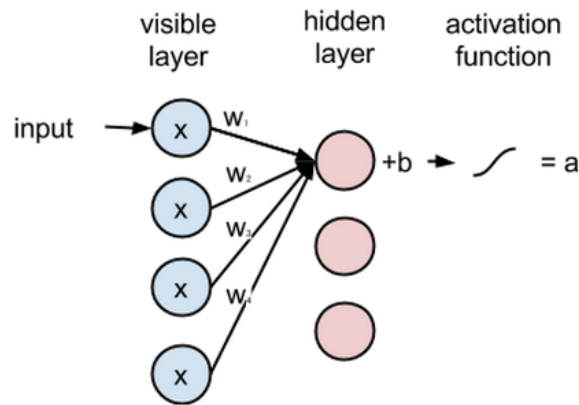
3.1.4 Βάρη

Όλοι οι νευρώνες του Hidden Layer ενώνονται με όλους τους νευρώνες του Visible Layer. Η κάθε ένωση νευρώνων είναι bidirectional και χαρακτηρίζεται από ένα βάρος, το οποίο έχει μια δεκαδική τιμή από 0 μέχρι 1. Η

τιμή του κάθε βάρους αλλάζει μετά από κάθε εποχή ανάλογα με το reconstruction error και τον ρυθμό μάθησης. Όλοι οι νευρώνες ενώνονται με όλους τους άλλους νευρώνες εκτός και αν ανήκουν στο ίδιο layer. Αυτό σημαίνει ότι εάν έχουμε για παράδειγμα X νευρώνες στο φανερό επίπεδο και Y στο κρυφό επίπεδο, τότε κάθε νευρώνας στο κρυφό επίπεδο παίρνει X βάρη από τους νευρώνες του φανερού επιπέδου, και κάθε νευρώνας στο φανερό επίπεδο παίρνει Y βάρη από τους νευρώνες του κρυφού επιπέδου. Άρα συνολικά έχουμε $X * Y$ βάρη.

3.1.5 Έξοδος Νευρώνων

Η έξοδος του κάθε νευρώνα έχει δυαδικό αποτέλεσμα. Η έξοδος του νευρώνα προκύπτει από το αποτέλεσμα της σιγμοειδούς συνάρτησης στην οποία δίνεται ως είσοδος το αποτέλεσμα του εσωτερικού γινομένου για όλα τα εισερχόμενα βάρη προς τον νευρώνα με τα αντίστοιχα δεδομένα. Από το αποτέλεσμα του Dot product αφαιρείται το βάρος του κατωφλίου του κάθε νευρώνα αντίστοιχα. Οι νευρώνες του RBM χαρακτηρίζονται και στοχαστικοί για τον λόγο ότι το τελικό αποτέλεσμα χαρακτηρίζεται από μία πιθανότητα. Η τελική έξοδος δίνεται από έναν τυχαίο αριθμό που παράγεται στιγμιαία. Εάν το αποτέλεσμα της σιγμοειδούς συνάρτησης είναι μεγαλύτερο από τον τυχαίο αριθμό, τότε η έξοδος γίνεται 1 και τότε λέμε ότι ο νευρώνας ενεργοποιήθηκε. Αλλιώς το αποτέλεσμα μένει μηδέν και λέμε ότι ο νευρώνας παρέμεινε ανενεργός. Στην εικόνα 3.1.5.1 μπορούμε να δούμε τα πιο πάνω σε γραφική μορφή.



Εικόνα 3.1.5.1 [20] Εδώ φαίνεται η γραφική αναπαράσταση του δικτύου.

3.1.6 Αλγόριθμος Εκπαίδευσης.

Το δίκτυο εκπαιδεύεται με τον αλγόριθμο Contrastive Divergence (Hinton 2002[15]). Ο αλγόριθμος είναι απλός. Κάθε δεδομένο εισόδου είναι ένα vector που αντιπροσωπεύει την πληροφορία αυτή κωδικοποιημένη με τέτοιο τρόπο που να μπορεί να γίνει η εκπαίδευση. Λόγω των περιορισμών που υπάρχουν εξ ορισμού στις ενώσεις των νευρώνων του RBM ο αλγόριθμος μπορεί να εκτελεστεί παράλληλα. Γι' αυτό είναι πρακτικό τα δεδομένα να φορτώνονται σε vectors. Κάθε στοιχείο του vector ανατίθεται σε ένα νευρώνα του visible layer για να γίνει εισαγωγή στο δίκτυο. Έπειτα, παράλληλα υπολογίζεται το εσωτερικό γινόμενο για κάθε νευρώνα του Hidden Layer. Το αποτέλεσμα του εσωτερικού γινομένου αφαιρείται από το αντίστοιχο κατώφλι (bias) του κάθε νευρώνα και δίνεται σαν είσοδος στην σιγμοειδή συνάρτηση και δίνει ένα αποτέλεσμα από το 0 μέχρι το 1, το αποτέλεσμα το οποίο ονομάζεται και ως η πιθανότητα του νευρώνα για να γίνει ενεργός. Γι' αυτό τον λόγο οι νευρώνες του Restricted Boltzmann machine ονομάζονται και στοχαστικοί. Συγκεκριμένα, μετά από το αποτέλεσμα τις σιγμοειδούς, παράγεται ένας τυχαίος αριθμός για κάθε νευρώνα. Εάν το αποτέλεσμα τις σιγμοειδούς είναι μεγαλύτερο από τον τυχαίο αριθμό, τότε η έξοδος του νευρώνα γίνεται 1 και λέμε ότι ο νευρώνας έγινε ενεργός. Το αντίθετο συμβαίνει σε αντίθετη περίπτωση. Η διαδικασία που περιεγράφηκε μέχρι τώρα ονομάζεται positive phase. Μετά από το positive phase, ακολουθεί το negative phase. Κατά το negative phase, ακολουθείται η ίδια διαδικασία με την διαφορά ότι η φορά των βαρών είναι η αντίθετη και αντί

για δεδομένα εκπαίδευσης στην είσοδο, έχουμε τις εξόδους του Hidden Layer. Στο positive phase ο αλγόριθμος κατευθυνόταν από το visible layer προς το hidden layer με σκοπό να βρούμε την έξοδο των νευρώνων του hidden layer. Στο negative phase υπολογίζεται με τον ίδιο τρόπο η έξοδος των νευρώνων του visible layer. Όλο αυτό γίνεται για να υπολογιστεί το reconstruction error και ως εκ τούτου να γίνουν update τα βάρη του δικτύου με την εξίσωση στην εικόνα 3.1.6.1. Η positive και η negative φάση μπορούν να ονομαστούν ως και 1 step Gibbs sampling και ο αλγόριθμος ως CD-1. Η πιο πάνω διαδικασία μπορεί να εφαρμοστεί και περισσότερες φορές, όμως κατά την φάση εκπαίδευσης ένα βήμα είναι ικανοποιητικό. Αυτό επιτρέπεται λόγω και του γεγονότος ότι οι συνδέσεις είναι bidirectional. Επαναλαμβάνεται η πιο πάνω διαδικασία για όλα τα δεδομένα και όλες τις εποχές εκπαίδευσης και μετά ακολουθεί η φάση επαλήθευσης (δοκιμής). Κατά αυτή την φάση όλες οι εικόνες δοκιμής γίνονται sampling στο εκπαιδευμένο πλέον δίκτυο RBM. Η φάση δοκιμής είναι πανομοιότυπη με την φάση εκπαίδευσης με την μόνη διαφορά ότι τα βάρη δεν αλλάζουν και ότι κατά την φάση αυτή εφαρμόζονται αρκετές περισσότερες επαναλήψεις Gibbs Sampling. Η έξοδος των νευρώνων του hidden Layer κατά το τελευταίο βήμα του Gibbs Sampling είναι και η τελική έξοδος του δικτύου.

$$\Delta w_{ij} = \epsilon (\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon})$$

Εικόνα 3.1.6.1 Εδώ φαίνεται η εξίσωση με την οποία τα βάρη του δικτύου υπολογίζονται.

3.1.7 Το δίκτυο από Μαθηματική/Φυσική σκοπιά.

Όπως περιεγράφηκε πιο πάνω υπολογίζουμε το reconstruction error με σκοπό να κάνουμε update τα βάρη. Ο λόγος που γίνεται το πιο πάνω είναι για να καταφέρουμε να δώσουμε στα βάρη τέτοιες τιμές, με σκοπό να μπορούν να απομνημονεύσουν την σωστή πληροφορία που τους δίνεται στην εκπαίδευση, έτσι ώστε να μπορούν να συμπληρώσουν την αποκρυμμένη πληροφορία στην φάση δοκιμής. Σύμφωνα με την θεωρία μεταξύ των 2 layers κρύβεται μια

ενέργεια η οποία δίνεται από τον τύπο που φαίνεται στην εικόνα 3.1.7.1. Στόχος της εκπαίδευσης είναι ανάλογα με κάθε δεδομένο εισόδου, το δίκτυο να βγάξει υψηλή ενέργεια στους νευρώνες που όντως θα έπρεπε να ανιχνεύουν κάτι. Στην προκειμένη περίπτωση, με τις εικόνες προσώπων, εάν σε ένα δεδομένο εισόδου το δίκτυο βγάξει αποτέλεσμα 1 σε κάποιο νευρώνα το οποίο αντίστοιχα στο δεδομένο εισόδου είναι 0, τότε τα βάρη που συνδέονται με αυτόν τον νευρώνα μειώνονται. Σε αντίθετη περίπτωση αυξάνονται. Σε γενικές γραμμές οι νευρώνες που βγάζουν 1 ως έξοδο σημαίνει ότι κάτι ανιχνεύσαν, και στην προκειμένη περίπτωση με τις μαυρόασπρες εικόνες έχουμε ένα μαύρο εικονοστοιχείο. Εάν κατά την φάση εκπαίδευσης το δίκτυο κάνει λάθος σε ένα εικονοστοιχείο τότε αυτόματα τα βάρη που ενώνονται με τον νευρώνα που αντιπροσωπεύει αυτό το εικονοστοιχείο πρέπει να αλλάξουν αναλόγως με την επιθυμητή έξοδο.

$$E(\mathbf{v}, \mathbf{h}) = - \sum_{i \in \text{visible}} a_i v_i - \sum_{j \in \text{hidden}} b_j h_j - \sum_{i,j} v_i h_j w_{ij}$$

Εικόνα 3.1.7.1 Εδώ φαίνεται η εξίσωση με την οποία υπολογίζεται η ενέργεια που κρύβεται μεταξύ hidden και visible layer.

Το δίκτυο δίνει μια πιθανότητα σε κάθε πιθανό συνδυασμό visible και hidden vector από την εξίσωση που φαίνεται στην εικόνα 3.1.7.2

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v}, \mathbf{h})}$$

Εικόνα 3.1.7.2 Εδώ φαίνεται η εξίσωση πιθανότητας που δίνει το δίκτυο σε οποιοδήποτε συνδυασμό visible και hidden vector. Το Z ονομάζεται partition function και είναι το άθροισμα των ενεργειών όλων των πιθανών συνδυασμών visible και hidden vector.

Η πιθανότητα το δίκτυο να συγκλίνει σε κάποια εικόνα εισόδου μπορεί να αυξηθεί με την προσαρμογή των βαρών έτσι ώστε να αυξάνονται οι πιθανότητες που περιεγραφήκαν πιο πάνω αυτού του vector και να μειώνονται των υπόλοιπων που δεν έχουν κάποια σχέση. Αυτό θα έχει ως αποτέλεσμα να μειώνεται η πιθανότητα αυτών που έχουν χαμηλή ενέργεια και από την άλλη να αυξάνεται αυτών που έχουν υψηλή ενέργεια. Συνοψίζοντας, αυτά που έχουν μικρή ενέργεια δεν κάνουν αίσθηση στην συνάρτηση Z (partition function). Το αντίθετο συμβαίνει σε αυτά με μεγάλη ενέργεια.

Κεφάλαιο 4

Σχεδιασμός και Υλοποίηση Νευρωνικών Δικτύων

4.1 Προ επεξεργασία δεδομένων ελέγχου

4.2 Στατιστικές μέθοδοι για υπολογισμό σφάλματος

4.1 Προ επεξεργασία δεδομένων ελέγχου

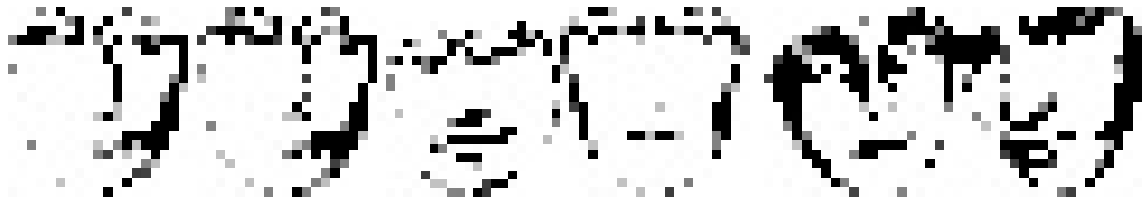
Οι εικόνες που είχαμε στην διάθεση μας θα έπρεπε να επεξεργαστούν με τέτοιο τρόπο έτσι ώστε να μπορούν να γίνουν κατανοητές από το δίκτυο, με απώτερο σκοπό το δίκτυο να αναγνωρίσει τα χαρακτηριστικά των εικόνων και στην φάση δοκιμής να μπορεί να συμπληρώνει οποιαδήποτε εικόνα. Οι εικόνες του κ. Λανίτη[14] που χρησιμοποιήσαμε στην κανονική τους μορφή ήταν σε μορφή grayscale. Οι grayscale εικόνες έχουν για κάθε εικονοστοιχείο μια τιμή φωτεινότητας από 0 μέχρι 255. Οι νευρώνες του RBM είναι binary, δηλαδή η έξοδος μας θα είναι 0 ή 1. Αυτό σημαίνει ότι δεν θα μπορούσαμε να βάζουμε ως είσοδο κανονικοποιημένα δεδομένα που αντιπροσωπεύαν 255 τιμές διότι μετά δεν θα μπορούσαμε να δούμε τα αποτελέσματα του δικτύου μας σε εικόνες. Αυτό είχε σαν αποτέλεσμα την μετατροπή των εικόνων του κ. Λανίτη[14] σε δυαδικές. Οι δυαδικές εικόνες είχαν μόνο 2 τιμές. 0 και 255, δηλαδή άσπρο και μαύρο. Το μόνο που έπρεπε να κάνουμε είναι να επιλέξουμε σωστά το κατώφλι και κατά το διάβασμα των δεδομένων να μετατρέψουμε όλα τα 255 σε 1.

Έπειτα, αλλάξαμε τις διαστάσεις των εικόνων και τις κάναμε να έχουν τετραγωνικές διαστάσεις. Οι διαστάσεις που επιλέξαμε να κάνουμε τις εικόνες είναι 20x20 για μείωση της πολυπλοκότητας και ως αποτέλεσμα αυτού την επιτάχυνση της εκπαίδευσης. Κάθε εικονοστοιχείο της εικόνας θα ήταν και ένας νευρώνας στο κάθε layer. Εάν επιλέγαμε εικόνες μεγάλων διαστάσεων θα

είχαμε μεγάλο αριθμό νευρώνων και κατά συνέπεια θα χρειαζόταν πολύ μεγαλύτερη υπολογιστική ισχύς και περισσότερος χρόνος στην εκπαίδευση του δικτύου. Από την στιγμή που το δίκτυο θα μπορούσε να απομνημονεύσει τις λεπτομέρειες από τις εικόνες και σε μικρές εικόνες τότε δεν υπήρχε κανένας λόγος να χρησιμοποιηθούν μεγαλύτερες εικόνες. Σε αυτό το σημείο ήρθαμε αντιμέτωποι με ένα μεγάλο δίλημμα. Οι εικόνες μεγαλύτερων διαστάσεων είχαν μεγαλύτερη ευκρίνεια κάτι που θα έκανε τα αποτελέσματα μας να είναι υψηλής ποιότητας. Από την άλλη όσο μεγάλωναν οι εικόνες τόσοος περισσότερος χρόνος θα χρειαζόταν κατά την διάρκεια της εκπαίδευσης. Δοκιμάσαμε να εκπαιδεύσουμε το δίκτυο με εικόνες 50 x 50 κάτι που είχε ως αποτέλεσμα το πρόγραμμα να καταναλώνει τεράστιους πόρους στον εξυπηρετητή που το τρέχαμε και να είναι πάρα πολύ αργό και ως συνέπεια αυτού να μην μπορούμε να πάρουμε αποτελέσματα πριν από την λήξη της εκπόνησης της ατομικής διπλωματικής μου εργασίας. Έτσι υποχρεωτικά καταλήξαμε στο γεγονός ότι θα εργαστούμε με εικόνες διαστάσεων 20 x 20. Κατά την διαδικασία μετατροπής των εικόνων σε binary μερικές εικόνες που παράγονταν δεν είχαν κάποιο νόημα και έτσι από τις 11022 εικόνες εργαστήκαμε με μόνο 1800 εικόνες. Οι υπόλοιπες είτε είχαν πολύ λίγη πληροφορία είτε είχαν πάρα πολύ χαμηλό φωτισμό και σαν αποτέλεσμα η τελική binary εικόνα είχε αρκετά μαύρα εικονοστοιχεία. Από τις τελικές εικόνες επιλέξαμε μερικές για εικόνες δοκιμής. Στις εικόνες που ακολουθούν μπορούμε να δούμε τις τελικές εικόνες καθώς και τις εικόνες που αποκλείσαμε από το training set μας.

Αξίζει να σημειωθεί ότι προσπαθήσαμε δύο φορές να παράξουμε δυαδικές εικόνες. Το πρώτο set από δυαδικές εικόνες που δημιουργήσαμε προσπαθήσαμε να κρατήσουμε εικόνες με όσο λιγότερα μαύρα εικονοστοιχεία αλλά ταυτοχρόνως να κρατήσουμε όσο το δυνατό περισσότερη λεπτομέρεια. Το δεύτερο set που αναγκαστήκαμε να δημιουργήσαμε είχαμε σαν πολιτική μας να μην ασχοληθούμε με τα μαύρα εικονοστοιχεία αλλά με τις λεπτομέρειες των εικόνων. Έτσι ακολουθήσαμε άλλη τακτική στην κατωφλίωση και επιλέγαμε το ανάλογο κατώφλι για την κάθε εικόνα ξεχωριστά ανάλογα με το ιστόγραμμα της. Έτσι τα αποτελέσματα των εικόνων ήταν και πιο ποιοτικά. Ο λόγος που διαλέξαμε να δημιουργήσουμε και μια δεύτερη συλλογή από εικόνες είναι τα αποτελέσματα του πρώτου set. Παρατηρήσαμε ότι τα δίκτυα που εκπαιδεύσαμε

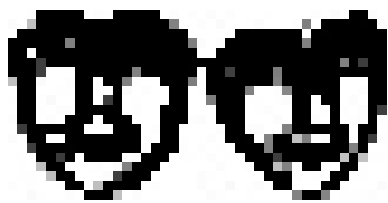
πάνω σε αυτές τις εικόνες δεν συμπλήρωναν μόνο την αποκρυμμένη πληροφορία αλλά και άλλες πληροφορίες. Έτσι φτάσαμε στο συμπέρασμα ότι οι εικόνες εκπαίδευσης που χρησιμοποιούσαμε μέχρι στιγμής ήταν πού χαμηλής ποιότητας και αυτό μερικές φορές έκανε τα δίκτυα μας να μπερδεύονται τόσο κατά την φάση της εκπαίδευσης αλλά και κατά την φάση της δοκιμής.



Εικόνα 4.1.1 Πιο πάνω βλέπουμε μερικές εικόνες από το πρώτο μας set. Οι πιο πάνω εικόνες είναι μερικές από τις 1400 εικόνες που επιλέξαμε να κρατήσουμε στο training set. Σε αντίθεση με τις εικόνες στην εικόνα 2.2.4 αυτές οι εικόνες έχουν νόημα.



Εικόνα 4.1.2 Πιο πάνω βλέπουμε μερικές εικόνες από το δεύτερο μας set. Οι πιο πάνω εικόνες είναι μερικές από τις 1900 εικόνες που επιλέξαμε να κρατήσουμε στο training set. Όπως βλέπουμε είναι καλύτερης ποιότητας από τις πιο πάνω.



Εικόνα 4.1.3 Πιο πάνω βλέπουμε μερικές φωτογραφίες από τις τελικές εικόνες που προέκυψαν μετά από την δυαδική επεξεργασία. Όπως μπορούμε εύκολα να παρατηρήσουμε οι φωτεινότητες των εικονοστοιχείων ήταν αρκετά χαμηλές και μετά την κατωφλίωση προέκυψαν αρκετά μαύρα εικονοστοιχεία, κάτι που μας οδήγησε στο να αφαιρέσουμε τέτοιου είδους εικόνες από τις εικόνες δοκιμής.

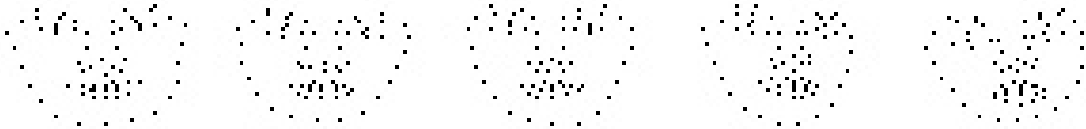
Σε αυτό το σημείο πρέπει να σχολιάσουμε και να δείξουμε τι εννοούμε όταν αναφέραμε πριν για το δίλημμα που είχαμε για το μέγεθος των εικόνων που θα χρησιμοποιούσαμε. Συγκεκριμένα όσο μικραίνει το μέγεθος των εικόνων τόσο πιο χαμηλή ήταν η ποιότητα των δεδομένων μας και στην τελική το δίκτυο εκπαιδευόταν πιο γρήγορα με λιγότερο ρεαλιστικά αποτελέσματα. Από την άλλη ισχύουν ακριβώς τα αντίθετα. Σε καμία περίπτωση δεν θα προτιμούσαμε τα δεδομένα χαμηλής ποιότητας εάν δεν είχαμε τον περιορισμό του χρόνου και της υπολογιστικής ισχύς. Έτσι επιλέξαμε να εργαστούμε με όσο καλύτερης ποιότητας δεδομένα μπορούσαμε αλλά και ταυτόχρονα χαμηλών διαστάσεων. Στην εικόνα 4.1.4 μπορούμε να δούμε τις διαφορές μεταξύ δεδομένων χαμηλής και μεσαίας ποιότητας.



Εικόνα 4.1.4 Στην εικόνα μπορούμε να δούμε στην πρώτη γραμμή εικόνες 20 x 20 που ήταν και αυτές που χρησιμοποιήσαμε την δεύτερη φορά για να εκπαιδεύσουμε το δίκτυο μας. Στην δεύτερη γραμμή βλέπουμε εικόνες από το ίδιο set όμως με διπλάσιες διαστάσεις. Οι διαφορές είναι εμφανέστατες.

Έκτος από τις δύο πιο πάνω βάσεις που δημιουργήσαμε θέλαμε να δοκιμάσουμε και το δίκτυο μας σε πιο εύκολα δεδομένα για να δείξουμε ότι πραγματικά αντιλαμβάνεται και συμπληρώνει το occlusion. Γι' αυτό τον λόγο δημιουργήσαμε και μια βάση δεδομένων με εικόνες προσώπων όπου περιέχουν 68 σημεία που αντιπροσωπεύουν τα βασικά χαρακτηριστικά του προσώπου. Για να παράξουμε την πιο πάνω βάση από εικόνες λάβαμε μερικά αρχεία από τον κ.

Λανίτη που περιείχαν συντεταγμένες που η κάθε μια αντιπροσώπευε ένα σημείο κλειδί. Στην εικόνα που ακολουθεί μπορούμε να δούμε μερικές από τις εικόνες που δημιουργήσαμε από τα αρχεία αυτά.



Εικόνα 4.1.5 Πιο πάνω βλέπουμε τις φωτογραφίες από την βάση δεδομένων που δημιουργήσαμε από τα αρχεία του κ. Λανίτη που περιείχαν συντεταγμένες των βασικών σημείων του κάθε προσώπου. Συγκεκριμένα η κάθε εικόνα περιείχε 68 σημεία.

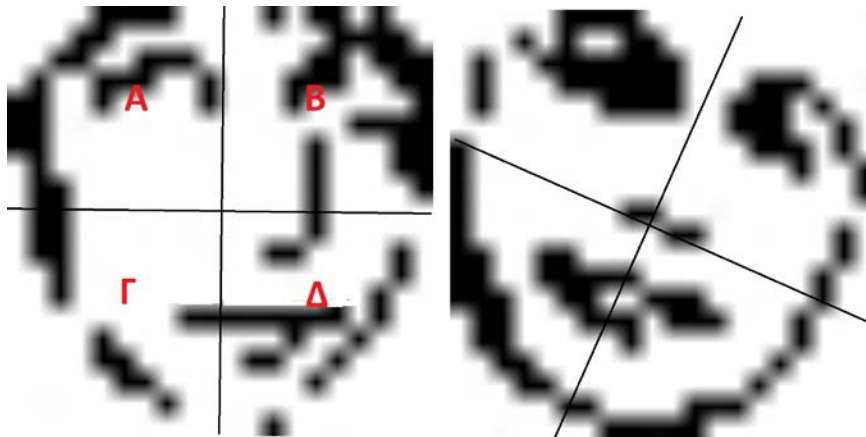
Επιπρόσθετα δοκιμάσαμε το δίκτυο μας και σε ένα πιο απλό πρόβλημα. Το πρόβλημα που επιλέξαμε ήταν το occlusion στους χειρόγραφους αριθμούς. Για το πιο πάνω πήραμε τα labelled δεδομένα από την βάση δεδομένων τις MNIST [19] με τους χειρόγραφους αριθμούς. Στην εικόνα 4.1.6 βλέπουμε μερικές από τις εικόνες της βάσης που περιγράψαμε πιο πάνω.



Εικόνα 4.1.6 Πιο πάνω βλέπουμε τις εικόνες από την βάση δεδομένων της MNIST με τους χειρόγραφους αριθμούς.

Σε αυτή την παράγραφο θα περιγράψει πλήρως ο τρόπος επεξεργασίας των δεδομένων ελέγχου - δοκιμής. Στα δεδομένα δοκιμής έγινε ακριβώς η ίδια διαδικασία που έγινε στα δεδομένα εκπαίδευσης με μια επιπλέον επεξεργασία η

οποία περιγράφεται πιο κάτω. Συγκεκριμένα έπρεπε να δημιουργήσουμε δεδομένα με αποκρυμμένη πληροφορία έτσι ώστε να μπορέσουμε να τα δώσουμε στο δίκτυο μετά την εκπαίδευση για να τα συμπληρώσει κατά την φάση δοκιμής. Αναλυτικά επιλέξαμε να χωρίσουμε την εικόνα σε 4 κομμάτια, όπως φαίνονται στην εικόνα 4.1.7. Επιλέξαμε τις εικόνες που θα δίνουμε στο δίκτυο για δοκιμή. Δημιουργήσαμε 3 αντίγραφα και επεξεργαστήκαμε το κάθε αντίγραφο ξεχωριστά. Στο πρώτο αντίγραφο αφαιρούσαμε οριζόντια πληροφορία, δηλαδή από δύο συνεχόμενα κομμάτια είτε τα 2 πάνω είτε τα 2 από κάτω. Στο δεύτερο αντίγραφο αφαιρούσαμε κάθετη πληροφορία δηλαδή από δύο τετράγωνα που βρίσκονταν στον ίδιο άξονα. Και τέλος από το τρίτο αντίγραφο αφαιρέσαμε διαγώνια πληροφορία είτε από την κύρια διαγώνιο είτε από την δευτερεύουσα διαγώνιο.



Εικόνα 4.1.7 Εδώ φαίνεται ο διαχωρισμός της εικόνας στα τέσσερα κομμάτια. Όπως μπορούμε να διακρίνουμε και στην εικόνα η αφαίρεση της κάθετης πληροφορίας μπορεί να είναι στα τετράγωνα ΑΓ ή στα τετράγωνα ΒΔ. Η οριζόντια στα ΑΒ ή ΓΔ και η διαγώνια στα τετράγωνα ΑΔ ή ΓΒ. Αξίζει να σημειωθεί ότι ο διαχωρισμός δεν γινόταν με βάση τις διαστάσεις των εικόνων αλλά με βάση την τοποθέτηση του προσώπου στην εικόνα. (βλ. 2^η εικόνα.)

4.2 Στατιστικές μέθοδοι για υπολογισμό σφάλματος

Στα RBMs χρησιμοποιείται το τετραγωνικό σφάλμα μεταξύ δεδομένων και reconstruction. Αυτό τον τρόπο χρησιμοποιήσαμε και εμείς στην υλοποίησή μας. Όμως, σύμφωνα με το άρθρο του κ. Hinton για τον αλγόριθμο Constructive

divergence το τετραγωνικό σφάλμα είναι πολύ φτωχή μέτρηση για την διαδικασία της μάθησης. Επιπρόσθετα στην παράγραφο για το μέσο τετραγωνικό σφάλμα, συστήνει την χρήση του όμως χωρίς να έχουμε εμπιστοσύνη σε αυτό. Δηλαδή είναι απλά μία ενδεικτική μέτρηση για την διαδικασία εκπαίδευσης, χωρίς όμως να αντιπροσωπεύσει απόλυτα την όλη διαδικασία εκπαίδευσης. Γι' αυτό τον λόγο έπρεπε να γίνουν διάφορες δοκιμές στο τελικό σφάλμα εκπαίδευσης. Με τα πιο πάνω εννοούμε ότι μπορεί το σφάλμα να φαίνεται χαμηλό, όμως αυτό εξαρτάται τόσο από τα δεδομένα εκπαίδευσης όσο και από την δυσκολία του προβλήματος. Θεωρητικά, εύκολα προβλήματα είναι πιο εύκολα στην εκμάθηση τους και γι' αυτό τον λόγο το τελικό σφάλμα αναμένεται να είναι χαμηλό. Αντιθέτως, πιο δύσκολα προβλήματα θα ήταν δύσκολα στην εκμάθηση και το σφάλμα εκπαίδευσης θα ήταν πιο υψηλό. Στην δεύτερη περίπτωση, μια λανθασμένη κίνηση ήταν να αναμένετε ένα πολύ χαμηλό σφάλμα, κάτι που θα οδηγούσε σε υπερ - εκπαίδευση του δικτύου με αποτέλεσμα την αδυναμία του για γενίκευση αφού αυτό που επιτεύχθηκε ήταν η «παπαγαλία» των δεδομένων εκπαίδευσης. Ο πιο πάνω τρόπος προσέγγισης θα ήταν ο βέλτιστος σε απλά προβλήματα όπως για παράδειγμα αυτό του exclusive or. Στην εικόνα 4.3.1 μπορούμε να δούμε την εξίσωση υπολογισμού του τετραγωνικού σφάλματος.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Εικόνα 4.2.1 Η εξίσωση υπολογισμού του τετραγωνικού σφάλματος.

Παράλληλα με το τετραγωνικό σφάλμα προσπαθήσαμε να βρούμε ένα ακόμη τρόπο για να μπορούμε και μαθηματικά να ελέγχουμε τα αποτελέσματα μας. Έτσι σκεφτήκαμε ένα αλγόριθμο ο οποίος θα μας έδινε μια μαθηματική ένδειξη για τα εικονοστοιχεία τα οποία το δίκτυο μας πρόσθεσε στην εικόνα με την αποκρυμμένη πληροφορία. Συγκεκριμένα, το επιθυμητό θα ήταν το δίκτυο να συμπληρώνει μόνο τα εικονοστοιχεία που απουσίαζαν. Αυτό γνωρίζαμε από την αρχή ότι ήταν αδύνατο για τον λόγο ότι τα δεδομένα εκπαίδευσης μας ήταν

σε τέτοια μορφή τα οποία σε μερικά σημεία είχαν ήδη occlusion και σε κάποια άλλα επιπλέον πληροφορία η οποία μπορεί να χαρακτηριστεί και ως θόρυβος. Άρα θα ήταν σχετικά αδύνατον ακόμη και μετά από τέλεια εκπαίδευση το δίκτυο μας να μην προσθέτει δεδομένα που δεν χρειάζονται. Γι' αυτό τον λόγο έπρεπε ο αλγόριθμος μας να μετρά μόνο τα εικονοστοιχεία στις ζητούμενες περιοχές. Ο αλγόριθμος είναι πολύ απλός. Ο αλγόριθμος μετρά την διαφορά των δύο εικόνων (εξόδου, πραγματικής) και εάν δύο αντίστοιχα εικονοστοιχεία τους έχουν διαφορετικό χρώμα, τότε η διαφορά αυξάνεται κατά ένα.

Κεφάλαιο 5

Αποτελέσματα και Συζήτηση

5.1 Τρόπος Αξιολόγησης αποτελεσμάτων

5.2 Αποτελέσματα MNIST Unlabeled Handwritten Digits

5.3 Αποτελέσματα training set 1

5.4 Αποτελέσματα training set 2

5.5 Αποτελέσματα με τις εικόνες σημείων

5.1 Τρόπος Αξιολόγησης αποτελεσμάτων

Όπως αναφέραμε και πιο πριν στο κεφάλαιο της προ επεξεργασίας των δεδομένων από τις greyscale εικόνες που είχαμε στην διάθεση μας, δημιουργήσαμε 2 training sets. Αναλυτικά στο πρώτο κρατήσαμε λιγότερες λεπτομέρειες για πιο εύκολη εκπαίδευση και στο δεύτερο κρατήσαμε εικόνες με μεγαλύτερη λεπτομέρεια με την ελπίδα για πιο ποιοτικά αποτελέσματα. Δοκιμάσαμε το δίκτυο μας σε διάφορες παραμέτρους. Οι παράμετροι που δοκιμάζαμε ήταν το τελικό τετραγωνικό σφάλμα για το οποίο σταματούσε η εκπαίδευση, ο ρυθμός μάθησης, και τα βήματα Gibbs sampling που θα εφαρμόζονταν κατά την φάση της δοκιμής. Μετά από πειραματισμό καταλήξαμε ότι τα ιδανικά βήματα Gibbs sampling ήταν 10000.

Ο τρόπος αξιολόγησης της εκπαίδευσης του δικτύου μας χωρίζεται σε δύο κατηγορίες. Η πρώτη κατηγορία είναι η ποσοτική μέτρηση της εκπαίδευσης και είναι το μέσο τετραγωνικό σφάλμα καθώς και η διαφορά εικονοστοιχείων μεταξύ πραγματικής και εικόνας εξόδου. Η δεύτερη κατηγορία είναι η ποιοτική

αξιολόγηση των αποτελεσμάτων. Με αυτό τον τρόπο αξιολόγησης δεν έχουμε κάποια μαθηματική ένδειξη για τα αποτελέσματα, παρά μόνο τα ίδια τα αποτελέσματα που είναι οι εικόνες που συμπλήρωσε το δίκτυο μας κατά την φάση δοκιμής. Πιο κάτω θα παρουσιάσουμε τα αποτελέσματα που πήραμε από την εκπαίδευση του δικτύου μας στην βάση δεδομένων της MNIST με χειρόγραφους αριθμούς.

5.2 Αποτελέσματα MNIST Unlabeled Handwritten Digits

Όπως είδαμε και στο κεφάλαιο με τα δεδομένα η βάση δεδομένων της MNIST περιέχει 60000 εικόνες εκπαίδευσης. Αυτή η βάση δεδομένων χρησιμοποιήθηκε αρκετά για την παράγωγή φίλτρων και για κατηγοριοποίηση εικόνων σε αρκετές εργασίες στο παρελθόν. Οι εικόνες αυτές είναι κωδικοποιημένες με ένα label στην αρχή της κάθε εικόνας πριν από το πρώτο byte δεδομένων. Εμείς πήραμε την βάση και την φέραμε στα μέτρα του δικού μας προβλήματος μας, έτσι αφαιρέσαμε το label στην αρχή. Ήταν μια αρχική δοκιμή του προβλήματος μας σε ένα πιο εύκολο πρόβλημα. Τα αποτελέσματα έδειξαν ότι το δίκτυο μας τα πήγε αρκετά καλά.

Πραγματικές Εικόνες:



Occluded Εικόνες:



Αποτελέσματα δικτύου:



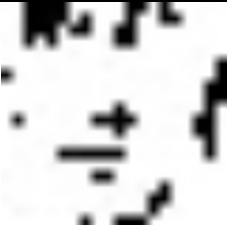
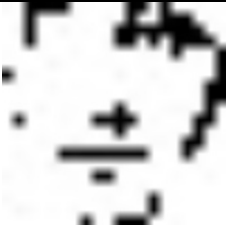
Σχολιασμός:

Για αυτά τα αποτελέσματα μπορούμε να πούμε ότι είναι αρκετά ικανοποιητικά. Αν πάρουμε τα αποτελέσματα από την αρχή μπορούμε να παρατηρήσουμε ότι στην εικόνα του αριθμού 9 το δίκτυο μας δεν συμπλήρωσε το occlusion και μετακίνησε και ένα κομμάτι τις εικόνες μερικά χιλιοστά πιο κάτω, όμως το αποτέλεσμα είναι κατανοητό. Στην εικόνα του αριθμού 8 συμπληρώνει επιτυχώς το κομμάτι που απουσιάζει προσθέτοντας ελάχιστο θόρυβο στην άκρια. Για τις επόμενες 3 εικόνες που ακολουθούν παρατηρούμε ότι συμπληρώνει τις εικόνες σωστά αλλάζοντας μερικά χαρακτηριστικά από την κάθε εικόνα. Συγκεκριμένα στην εικόνα του αριθμού 7 δίνει μια μικρή κλίση στην κάθετη γραμμή του. Στην εικόνα του αριθμού 6 αλλάζει το σχήμα του κύκλου ελαφρώς και στην εικόνα του αριθμού 5 κάνει πιο καμπυλωτή την κάτω ευθεία. Στην εικόνα του αριθμού 4 και του αριθμού 3 παρατηρούμε ότι η πραγματική εικόνα με την τελική έχουν σχεδόν καθόλου διαφορές. Και τέλος οι τελευταίες 3 εικόνες συμπληρώνονται σωστά. Παρατηρούμε επίσης ότι το δίκτυο χοντραίνει λίγο όλες τις γραμμές. Στις σελίδες που ακολουθούν παρουσιάζονται τα αποτελέσματα που πήραμε από το πρώτο training set με διαφορετικό τελικό τετραγωνικό σφάλμα.

5.3 Αποτελέσματα training set 1

Όπως είναι εμφανές από τις εικόνες, ακόμη και οι πραγματικές εικόνες είχαν μέσα θορύβους και αποκομμένα δεδομένα (occlusion). Αυτό είναι κάτι ανεπιθύμητο αφού με αυτό τον τρόπο δεν μπορούσαμε να εκπαιδεύσουμε το δίκτυο όπως εμείς θα θέλαμε. Με το πιο πάνω εννοούμε ότι εάν όλες μας οι εικόνες στην φάση εκπαίδευσης ήταν τέλειες, τότε αυτό θα σήμαινε ότι το δίκτυο θα εντόπιζε πολύ πιο εύκολα τις λεπτομέρειες των εικόνων.

Πείραμα 1^ο:

ΠΡΑΓΜΑΤΙΚΗ ΕΙΚΟΝΑ	ΚΑΘΕΤΟ OCCLUSION	ΔΙΑΓΩΝΙΟ OCCLUSION	ΟΡΙΖΟΝΤΙΟ OCCLUSION
ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΙΚΤΥΟΥ RBM ΜΕ ΤΕΛΙΚΟ ΤΕΤΡΑΓΩΝΙΚΟ ΣΦΑΛΜΑ 0.1 ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΙΚΤΥΟΥ RBM ΜΕ ΤΕΛΙΚΟ ΤΕΤΡΑΓΩΝΙΚΟ ΣΦΑΛΜΑ 0.01			
			
			

Σχολιασμός:

Αρχικά πρέπει να αναφερθεί ότι στην πραγματική-αρχική εικόνα προσώπου, υπήρχε ήδη occlusion πληροφορίας στην αριστερή πλευρά του περιγράμματος. Άρα θεωρητικά ήταν μια δύσκολη εικόνα για το δίκτυο δεδομένου ότι προσθέσαμε επιπλέον occlusion στην εικόνα.

Στην εικόνα με κάθετο occlusion αποκρύψαμε λεπτομέρεια από το δεξί μάτι και από την κάτω δεξιά πλευρά του περιγράμματος. Παρατηρούμε ότι το δίκτυο με τελικό σφάλμα 0.01 συμπληρώνει σε μεγάλο βαθμό το δεξί μάτι και πλήρως το αποκομμένο δεξί περίγραμμα. Παρατηρούμε επίσης ότι δίκτυο δεν συμπληρώνει το αριστερό occlusion που υπήρχε από πριν στην εικόνα. Στην αντίστοιχη εικόνα του δικτύου με τελικό σφάλμα 0.1, το δίκτυο συμπληρώνει το περίγραμμα όμως με μια μικρή δόση θορύβου. Επίσης, παρατηρούμε ότι

αλλάζει λανθασμένα εικονοστοιχεία στην εικόνα και το τελικό πρόσωπο έχει διαφορετικό σχήμα από το αρχικό. Αυτό είναι κάτι που είναι πλήρως ανεπιθύμητο. Στον πίνακα 5.1.1 μπορούμε και στατιστικά να δούμε ότι διαφορά είναι μεγάλη αφού μετατοπίζει ένα μεγάλο ποσοστό των εικονοστοιχείων.

Στην εικόνα του διαγώνιου occlusion αφαιρέσαμε λεπτομέρεια από το αριστερό μάτι και από το δεξί περίγραμμα. Παρόλο που υπήρχε ήδη occlusion στην αριστερή πλευρά του περιγράμματος επιλέξαμε να αφαιρέσουμε και από την δεξιά πλευρά για να δυσκολέψουμε το δίκτυο μας. Παρόλη την δυσκολία το δίκτυο με τελικό σφάλμα 0.01 τα πήγε περίφημα όπως μπορούμε να παρατηρήσουμε και από την συμπληρωμένη εικόνα. Συμπλήρωσε σχεδόν όλο το ήδη αποκομμένο περίγραμμα της πραγματικής εικόνας και επιπλέον σχημάτισε τέλεια το δεξί περίγραμμα και πρόσθεσε και τα απαραίτητα εικονοστοιχεία στο μάτι. Το δίκτυο με τελικό σφάλμα 0.1 συμπλήρωσε ικανοποιητικά το περίγραμμα με λίγο θόρυβο.

Τέλος, στην εικόνα με το οριζόντιο occlusion αφαιρέσαμε τα δύο μάτια. Το δίκτυο με τελικό σφάλμα 0.01 συμπλήρωσε τέλεια το δεξί και ελλιπώς το αριστερό μάτι. Επίσης, βλέπουμε ότι συμπληρώνει εικονοστοιχεία στην μύτη και στο αριστερό ήδη αποκομμένο περίγραμμα. Το δίκτυο με τελικό σφάλμα 0.1 συμπληρώνει τα μάτια πάρα πολύ καλά, εντούτοις αλλάζει το σχήμα του προσώπου και το κάνει πιο στρογγυλό όπως ακριβώς στην εικόνα με το κάθετο occasion. Αυτή η αλλαγή φαίνεται και στον πίνακα 5.3.1.

Γενικά για τα πιο πάνω αποτελέσματα μπορούμε να πούμε με σιγουριά ότι οι εικόνες του δικτύου που εκπαιδεύτηκε με τελικό σφάλμα 0.01 είναι αρκετά καλύτερες και πιο ευδιάκριτες.

Στατιστική Μελέτη Πειράματος 1:

	ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΚΑΘΕΤΟΥ OCCLUSION		ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΔΙΑΓΩΝΙΟΥ OCCLUSION		ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΟΡΙΖΟΝΤΙΟΥ OCCLUSION	
	Συνολικά	%	Συνολικά	%	Συνολικά	%
Τελικό Σφάλμα 0.1	87(41)	21.75	69(31)	17.25	82(49)	20.5
Τελικό Σφάλμα 0.01	65(31)	16.25	75(50)	18.75	82(45)	20.5

Πίνακας 5.3.1

***Στην παρένθεση είναι τα εικονοστοιχεία θορύβου (Μαύρα εικονοστοιχεία στην εικόνα εξόδου που δεν υπήρχαν στην πραγματική εικόνα).**

Σχολιασμός στατιστικής Μελέτης Πειράματος 1:

Παρατηρούμε από τον πιο πάνω πίνακα ότι στις περισσότερες περιπτώσεις τα λανθασμένα εικονοστοιχεία και ο θόρυβος μειώνονται στην εκτέλεση του δικτύου με τελικό σφάλμα 0.01. Επίσης από τις πιο πάνω μετρήσεις βλέπουμε δίπλα από τον αριθμό διαφοράς εικονοστοιχείων μέσα σε παρένθεση τα εικονοστοιχεία θορύβου. Παρατηρούμε ότι και στις δύο περιπτώσεις έχουμε υψηλό ποσοστό θορύβου. Με αυτό εννοούμε ότι εικονοστοιχεία της πραγματικής εικόνας δεν υπάρχουν στην εικόνα εξόδου.

Πείραμα 2°:

ΠΡΑΓΜΑΤΙΚΗ ΕΙΚΟΝΑ	ΚΑΘΕΤΟ OCCLUSION	ΔΙΑΓΩΝΙΟ OCCLUSION	ΟΡΙΖΟΝΤΙΟ OCCLUSION
			
ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΙΚΤΥΟΥ RBM ΜΕ ΤΕΛΙΚΟ ΤΕΤΡΑΓΩΝΙΚΟ ΣΦΑΛΜΑ 0.1			
ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΙΚΤΥΟΥ RBM ΜΕ ΤΕΛΙΚΟ ΤΕΤΡΑΓΩΝΙΚΟ ΣΦΑΛΜΑ 0.01			

Σχολιασμός:

Αρχικά πρέπει να αναφερθεί ότι η πραγματική-αρχική εικόνα προσώπου, ήταν πάρα πολύ καλή χωρίς επιπλέον θόρυβο και χωρίς occlusion. Άρα θεωρητικά ήταν μια εύκολη εικόνα για το δίκτυο.

Στην εικόνα με κάθετο occlusion αποκρύψαμε λεπτομέρεια από το δεξί μάτι και το περίγραμμα στην κάτω δεξιά μεριά. Παρατηρούμε ότι το δίκτυο με τελικό σφάλμα 0.01 συμπληρώνει αποτελεσματικά το δεξί μάτι και συμπληρώνει πλήρως το αποκομμένο περίγραμμα. Παρατηρούμε όμως ότι προσθέτει μερικά επιπλέον εικονοστοιχεία στην μύτη και χαλά το σχήμα της. Το δίκτυο με τελικό σφάλμα 0.1 συμπληρώνει το μάτι και το περίγραμμα σχετικά καλά. Παρατηρούμε ότι η εικόνα του δικτύου με τελικό σφάλμα 0.1 είναι πιο εμφανίσιμη και πιο κοντά στην πραγματική. Μπορούμε επιπλέον να

παρατηρήσουμε ότι η εικόνα αυτή είναι ακόμη καλύτερη από την πραγματική. Εάν παρατηρήσουμε την μύτη της συμπληρωμένης εικόνας προσώπου είναι αρκετά ρεαλιστική και πιο κοντά στην πραγματικότητα. Αυτό είναι ακόμη μια ένδειξη για τα δεδομένα εκπαίδευσης μας. Μπορούμε εύκολα να κατανοήσουμε ότι τα δεδομένα μας μετά την κατωφλίωση έχασαν αρκετή λεπτομέρεια κάτι που κάνει τα αποτελέσματα του δικτύου μας λιγότερο ρεαλιστικά.

Στην εικόνα του διαγώνιου occlusion αφαιρέσαμε λεπτομέρεια από το δεξί μάτι και από την κάτω αριστερή μεριά του περιγράμματος. Το δίκτυο με τελικό σφάλμα 0.01 τα πήγε περίφημα όπως μπορούμε να παρατηρήσουμε και από την συμπληρωμένη εικόνα. Συμπλήρωσε σχεδόν όλο το ήδη αποκομμένο περίγραμμα της πραγματικής εικόνας και επιπλέον πρόσθεσε τα απαραίτητα εικονοστοιχεία στο μάτι. Το δίκτυο με τελικό σφάλμα 0.1 συμπλήρωσε ελαφρώς το δεξί μάτι και πλήρως το περίγραμμα. Παρατηρούμε ότι στο συμπλήρωμα του περιγράμματος το δίκτυο μας πρόσθεσε και επιπλέον θόρυβο, και άλλαξε ελαφρώς την μορφή του στόματος, κάτι που όπως ξαναείπαμε είναι ανεπιθύμητο.

Τέλος, στην εικόνα με το οριζόντιο occlusion αφαιρέσαμε τα μάτια. Το δίκτυο με τελικό σφάλμα 0.01 συμπλήρωσε τέλεια τα μάτια. Το δίκτυο με τελικό σφάλμα 0.1 συμπλήρωσε και τα δύο μάτια ικανοποιητικά όμως πρόσθεσε εικονοστοιχεία στην μύτη που δεν υπάρχουν στην αρχική εικόνα.

Για τις πιο πάνω εικόνες μπορούμε να πούμε αυτό που αναφέρθηκε και πιο πάνω για τα δεδομένα εκπαίδευσης. Συγκεκριμένα σε δύο εικόνες το δίκτυο πρόσθεσε εικονοστοιχεία σε σημεία που δεν ήταν το ζητούμενο. Το οξύμωρο της υπόθεσης αυτής είναι ότι τα εικονοστοιχεία που πρόσθεσε έκαναν τις εικόνες πιο φωτορεαλιστικές ακόμη και από την αρχική εικόνα.

Στατιστική Μελέτη Πειράματος 2:

	ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΚΑΘΕΤΟΥ OCCLUSION		ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΔΙΑΓΩΝΙΟΥ OCCLUSION		ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΟΡΙΖΟΝΤΙΟΥ OCCLUSION	
	Συνολικά	%	Συνολικά	%	Συνολικά	%
Τελικό Σφάλμα 0.1	56(29)	14	54(30)	13.5	85(51)	21.25
Τελικό Σφάλμα 0.01	72(39)	18	71(30)	17.75	76(41)	19

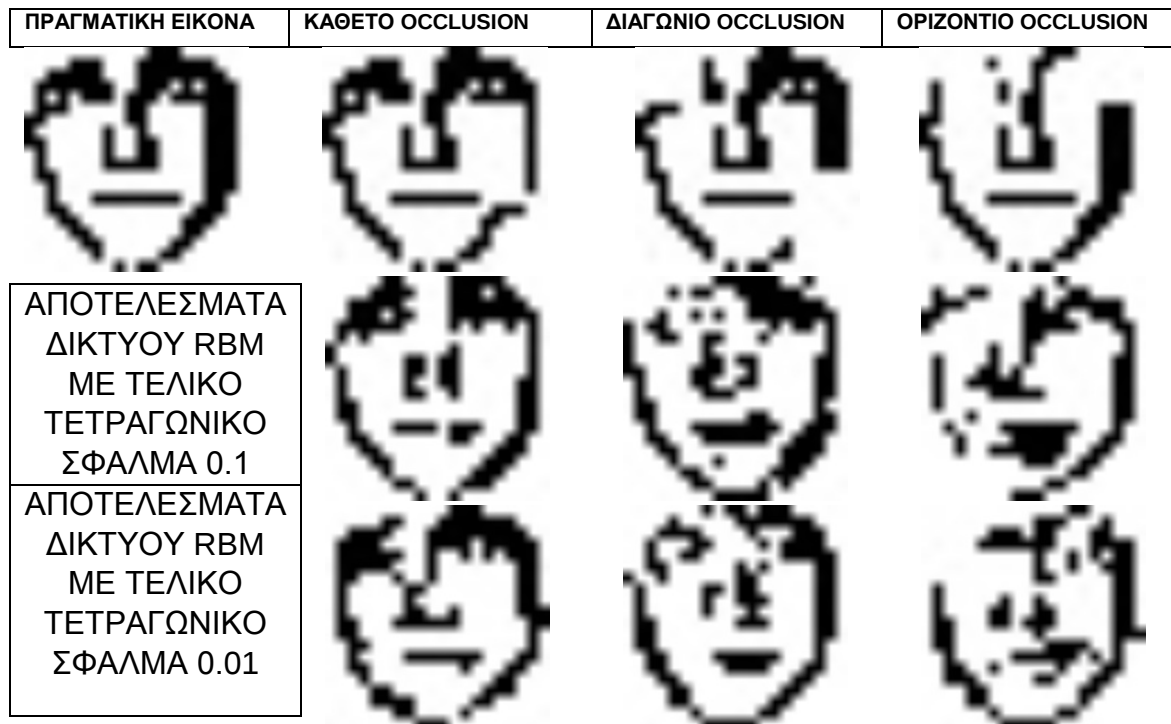
Πίνακας 5.3.2

*Στην παρένθεση είναι τα εικονοστοιχεία θορύβου (Μαύρα εικονοστοιχεία στην εικόνα εξόδου που δεν υπήρχαν στην πραγματική εικόνα).

Σχολιασμός στατιστικής Μελέτης Πειράματος 2:

Παρατηρούμε από τον πιο πάνω πίνακα ότι στις περισσότερες περιπτώσεις τα λανθασμένα εικονοστοιχεία και ο θόρυβος δεν μειώνονται στην εκτέλεση του δικτύου με τελικό σφάλμα 0.01. Επίσης από τις πιο πάνω μετρήσεις βλέπουμε δίπλα από τον αριθμό διαφοράς εικονοστοιχείων μέσα σε παρένθεση τα εικονοστοιχεία θορύβου. Παρατηρούμε ότι και στις δύο περιπτώσεις έχουμε υψηλό ποσοστό θορύβου. Με αυτό εννοούμε ότι εικονοστοιχεία της πραγματικής εικόνας δεν υπάρχουν στην εικόνα εξόδου. Τα λανθασμένα εικονοστοιχεία του δικτύου με τελικό σφάλμα 0.01 είναι περισσότερα.

Πείραμα 3°:



Σχολιασμός:

Αρχικά πρέπει να αναφερθεί ότι η πραγματική-αρχική εικόνα προσώπου, ήταν αρκετά καλή με ελαφρύ θόρυβο και χωρίς occlusion.

Στην εικόνα με κάθετο occlusion αποκρύψαμε λεπτομέρεια από το δεξί περίγραμμα. Παρατηρούμε ότι το δίκτυο με τελικό σφάλμα 0.01 συμπληρώνει αποτελεσματικά το αποκομμένο περίγραμμα. Παρατηρούμε επίσης ότι προσθέτει και άλλα εικονοστοιχεία στην εικόνα μερικά από τα οποία είναι αχρείαστα. Παρομοίως μπορούμε να σχολιάσουμε και την εικόνα του δικτύου με τελικό σφάλμα 0.1.

Στην εικόνα του διαγώνιου occlusion αφαιρέσαμε λεπτομέρεια από το αριστερό μάτι και από το δεξί περίγραμμα. Το δίκτυο με τελικό σφάλμα 0.01 τα πήγε αρκετά καλά στις ζητούμενες περιοχές όπως μπορούμε να παρατηρήσουμε και από την συμπληρωμένη εικόνα. Συμπλήρωσε σχεδόν όλο το ήδη αποκομμένο περίγραμμα της πραγματικής εικόνας και επιπλέον

πρόσθεσε τα απαραίτητα εικονοστοιχεία στο μάτι. Το αρνητικό στην εικόνα αυτή είναι το γεγονός ότι προσθέτει εικονοστοιχεία που δεν είναι απαραίτητα και αλλάζει τις μορφές των λεπτομερειών. Η εικόνα του δικτύου με τελικό σφάλμα 0.1 είναι ελαφρώς χειρότερη.

Τέλος, στην εικόνα με το οριζόντιο occlusion αφαιρέσαμε τα μάτια. Το δίκτυο με τελικό σφάλμα 0.01 συμπλήρωσε ελαφρώς το δεξί μάτι όμως καθόλου το αριστερό. Τα ίδια σχόλια μπορούμε να πούμε και για την εικόνα του δικτύου με τελικό σφάλμα 0.1 με την προσθήκη ότι προσθέτει και άλλα αχρείαστα εικονοστοιχεία.

Μπορούμε να πούμε ότι σε αυτές τις εικόνες φαίνεται το δίκτυο να μπερδεύτηκε αρκετά σε αυτό που έπρεπε να κάνει με αποτέλεσμα να προσθέτει στοιχεία σε λανθασμένα σημεία.

Στατιστική Μελέτη Πειράματος 3:

	ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΚΑΘΕΤΟΥ OCCLUSION		ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΔΙΑΓΩΝΙΟΥ OCCLUSION		ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΟΡΙΖΟΝΤΙΟΥ OCCLUSION	
	Συνολικά	%	Συνολικά	%	Συνολικά	%
Τελικό Σφάλμα 0.1	61(22)	15.25	85(39)	21.25	106(38)	26.5
Τελικό Σφάλμα 0.01	62(25)	15.5	71(24)	17.75	87(28)	21.75

Πίνακας 5.3.3

*Στην παρένθεση είναι τα εικονοστοιχεία θορύβου (Μαύρα εικονοστοιχεία στην εικόνα εξόδου που δεν υπήρχαν στην πραγματική εικόνα).

Σχολιασμός στατιστικής Μελέτης Πειράματος 3:

Παρατηρούμε από τον πιο πάνω πίνακα ότι στις περισσότερες περιπτώσεις τα λανθασμένα εικονοστοιχεία και ο θόρυβος μειώνονται στην εκτέλεση του δικτύου με τελικό σφάλμα 0.01 και συγκεκριμένα στις περιπτώσεις διαγώνιου και οριζόντιου occlusion. Επίσης από τις πιο πάνω μετρήσεις βλέπουμε δίπλα από τον αριθμό διαφοράς εικονοστοιχείων μέσα σε παρένθεση τα εικονοστοιχεία θορύβου. Παρατηρούμε ότι και στις δύο περιπτώσεις έχουμε

υψηλό ποσοστό θορύβου. Με αυτό εννοούμε ότι εικονοστοιχεία της πραγματικής εικόνας δεν υπάρχουν στην εικόνα εξόδου.

Πείραμα 4^ο:

ΠΡΑΓΜΑΤΙΚΗ ΕΙΚΟΝΑ	ΚΑΘΕΤΟ OCCLUSION	ΔΙΑΓΩΝΙΟ OCCLUSION	ΟΡΙΖΟΝΤΙΟ OCCLUSION
			
ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΙΚΤΥΟΥ RBM ΜΕ ΤΕΛΙΚΟ ΤΕΤΡΑΓΩΝΙΚΟ ΣΦΑΛΜΑ 0.1 ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΙΚΤΥΟΥ RBM ΜΕ ΤΕΛΙΚΟ ΤΕΤΡΑΓΩΝΙΚΟ ΣΦΑΛΜΑ 0.01			
			

Σχολιασμός:

Αρχικά πρέπει να αναφερθεί ότι η πραγματική-αρχική εικόνα προσώπου, ήταν αρκετά καλή με ελαφρύ θόρυβο στο στόμα και την μύτη και χωρίς occlusion. Εντούτοις τα δύο δίκτυα φάνηκε ότι δυσκολεύτηκαν αρκετά σε αυτή την εικόνα.

Στην εικόνα με κάθετο occlusion αποκρύψαμε λεπτομέρεια από την μύτη και το στόμα. Παρατηρούμε ότι το δίκτυο με τελικό σφάλμα 0.01 συμπληρώνει αποτελεσματικά το αποκομμένο στόμα όχι όμως την μύτη. Το δίκτυο με τελικό σφάλμα 0.1 δεν τα πήγε καθόλου καλά και το αποτέλεσμα δεν θυμίζει το πραγματικό και δεν είναι καθόλου ρεαλιστικό. Συγκεκριμένα άλλαξε εικονοστοιχεία που δεν έπρεπε.

Στην εικόνα του διαγώνιου occlusion αφαιρέσαμε λεπτομέρεια από το αριστερό μάτι και από το δεξί περίγραμμα. Το δίκτυο με τελικό σφάλμα 0.01 τα

πήγε αρκετά καλά όπως μπορούμε να παρατηρήσουμε και από την συμπληρωμένη εικόνα. Συμπλήρωσε σχεδόν όλο το ήδη αποκομμένο περίγραμμα της πραγματικής εικόνας και επιπλέον πρόσθεσε τα απαραίτητα εικονοστοιχεία στο μάτι. Το δίκτυο με τελικό σφάλμα 0.1 δεν τα πήγε το ίδιο καλά με το άλλο.

Τέλος, στην εικόνα με το οριζόντιο occlusion αφαιρέσαμε το στόμα. Το δίκτυο με τελικό σφάλμα 0.01 πρόσθεσε μερικά όμως όχι ικανοποιητικά εικονοστοιχεία στο στόμα. Το δίκτυο με τελικό σφάλμα 0.1 πρόσθεσε και αυτό μερικά εικονοστοιχεία στην μύτη όμως άλλαξε και άλλα που δεν έπρεπε κάνοντας την εικόνα αρκετά κακή.

Για τις πιο πάνω εικόνες μπορούμε να πούμε ότι το δίκτυο με τελικό σφάλμα 0.01 τα πήγε σχετικά καλά, όμως δεν μπορούμε να πούμε το ίδιο και για το άλλο δίκτυο. Φαίνεται ότι μπερδεύτηκε αρκετά σε αυτό που έπρεπε να κάνει με αποτέλεσμα να προσθέτει στοιχεία σε λανθασμένα σημεία.

Στατιστική Μελέτη Πειράματος 4:

	ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΚΑΘΕΤΟΥ OCCLUSION		ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΔΙΑΓΩΝΙΟΥ OCCLUSION		ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΟΡΙΖΟΝΤΙΟΥ OCCLUSION	
	Συνολικά	%	Συνολικά	%	Συνολικά	%
Τελικό Σφάλμα 0.1	91(35)	22.75	91(33)	22.75	91(23)	22.75
Τελικό Σφάλμα 0.01	74(27)	18.5	76(26)	19.5	86(27)	21.5

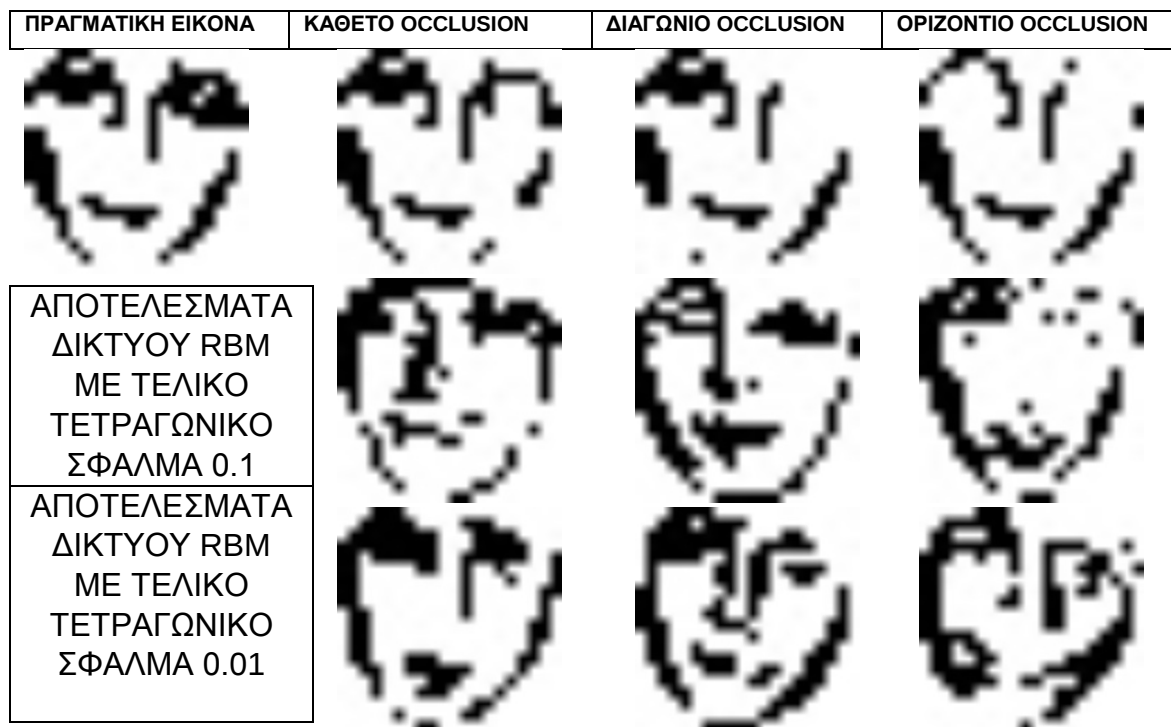
Πίνακας 5.3.4

***Στην παρένθεση είναι τα εικονοστοιχεία θορύβου (Μαύρα εικονοστοιχεία στην εικόνα εξόδου που δεν υπήρχαν στην πραγματική εικόνα).**

Σχολιασμός στατιστικής Μελέτης Πειράματος 4:

Παρατηρούμε από τον πιο πάνω πίνακα ότι τα λανθασμένα εικονοστοιχεία μειώνονται στην εκτέλεση του δικτύου με τελικό σφάλμα 0.01. Ο θόρυβος όμως παραμένει σε υψηλά ποσοστά.

Πείραμα 5°:



Σχολιασμός:

Αρχικά πρέπει να αναφερθεί ότι η πραγματική-αρχική εικόνα προσώπου, ήταν καλή με ελαφρύ θόρυβο και λίγο occlusion. Άρα θεωρητικά ήταν σχετικά εύκολη εικόνα για το δίκτυο. Αναμένονταν καλά αποτελέσματα τόσο στο κάθετο occlusion όσο και στο διαγώνιο.

Στην εικόνα με κάθετο occlusion αποκρύψαμε λεπτομέρεια από το κάτω μέρος του δεξιού περιγράμματος και απο το δεξί μάτι. Παρατηρούμε ότι το δίκτυο με τελικό σφάλμα 0.01 συμπληρώνει αποτελεσματικά το αποκομμένο περίγραμμα και το μάτι. Παρατηρούμε επίσης ότι προσθέτει και άλλα εικονοστοιχεία στην εικόνα και τείνει να συμπληρώσει και το κάτω κομμάτι της εικόνας που ήταν ήδη occluded από πριν. Αυτό είναι ένα αρκετά καλό αποτέλεσμα. Το δίκτυο με τελικό σφάλμα 0.1 παρατηρούμε τα ίδια αποτελέσματα με το άλλο δίκτυο με την διαφορά ότι σε αυτό προσθέτει εικονοστοιχεία που δεν χρειάζονται στην μύτη. Τα εικονοστοιχεία που πρόσθεσε όμως έκαναν την μύτη πιο φωτορεαλιστική.

Στην εικόνα του διαγώνιου occlusion αφαιρέσαμε λεπτομέρεια από το δεξί μάτι και από το κάτω αριστερό μέρος του περιγράμματος. Το δίκτυο με τελικό σφάλμα 0.01 τα πήγε αρκετά καλά όπως μπορούμε να παρατηρήσουμε και από την συμπληρωμένη εικόνα. Συμπλήρωσε σχεδόν όλο το ήδη αποκομμένο περίγραμμα της πραγματικής εικόνας και επιπλέον πρόσθεσε τα απαραίτητα εικονοστοιχεία στο μάτι. Επιπλέον μπορούμε να παρατηρήσουμε ότι πρόσθεσε εικονοστοιχεία στην μύτη και έκανε το αποτέλεσμα πιο φωτορεαλιστικό. Παρόμοιες παρατηρήσεις μπορούμε να πούμε και για την εικόνα του δικτύου με σφάλμα 0.1.

Τέλος, στην εικόνα με το οριζόντιο occlusion αφαιρέσαμε τα μάτια. Το δίκτυο με σφάλμα 0.01 συμπλήρωσε ελαφρώς τόσο το δεξί μάτι όσο και το αριστερό μάτι. Μπορούμε να πούμε ότι αυτή η εικόνα είναι σχετικά καλή αν τις συγκρίνουμε και με τις προηγούμενες εικόνες του διαγώνιου occlusion. Το δίκτυο με τελικό σφάλμα 0.1 έφτιαξε μια πολύ κακή εικόνα που είναι ανεπιθύμητη στα αποτελέσματα.

Ως γενική παρατήρηση για τα πιο πάνω μπορούμε να πούμε ότι το δίκτυο με σφάλμα 0.01 τα πήγε αρκετά καλά. Επίσης μπορούμε να θέσουμε ξανά την ίδια παρατήρηση για τα δεδομένα εκπαίδευσης και την τάση των δύο δικτύων να συμπληρώνουν τις εικόνες σε μη ζητούμενες περιοχές και εντούτοις αρκετές φορές να κάνουν το αποτέλεσμα πιο ρεαλιστικό.

Στατιστική Μελέτη Πειράματος 5:

	ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΚΑΘΕΤΟΥ OCCLUSION		ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΔΙΑΓΩΝΙΟΥ OCCLUSION		ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΟΡΙΖΟΝΤΙΟΥ OCCLUSION	
	Συνολικά	%	Συνολικά	%	Συνολικά	%
Τελικό Σφάλμα 0.1	96(48)	24	81(43)	20.25	124(56)	31
Τελικό Σφάλμα 0.01	58(31)	14.5	75(40)	18.75	93(41)	23.25

Πίνακας 5.3.5

*Στην παρένθεση είναι τα εικονοστοιχεία θορύβου (Μαύρα εικονοστοιχεία στην εικόνα εξόδου που δεν υπήρχαν στην πραγματική εικόνα).

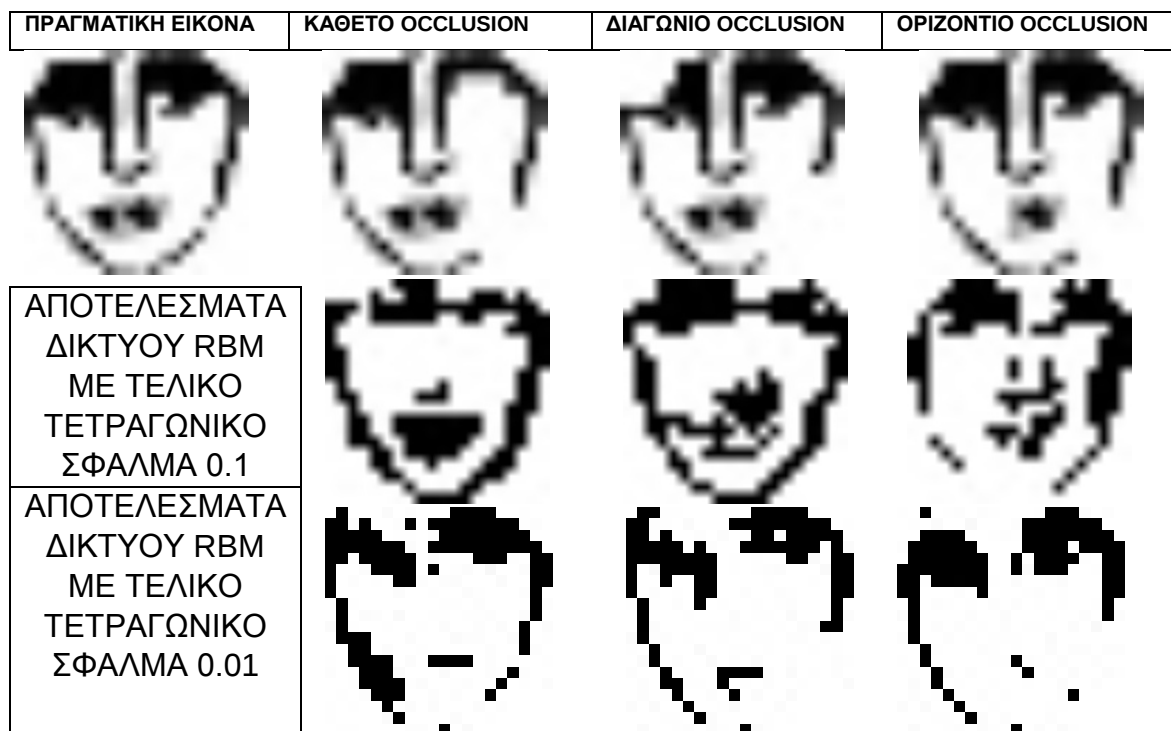
Σχολιασμός στατιστικής Μελέτης Πειράματος 5:

Παρατηρούμε από τον πιο πάνω πίνακα ότι τα λανθασμένα εικονοστοιχεία μειώνονται δραστικά στην εκτέλεση του δικτύου με τελικό σφάλμα 0.01. Επίσης από τις πιο πάνω μετρήσεις βλέπουμε δίπλα από τον αριθμό διαφοράς εικονοστοιχείων μέσα σε παρένθεση τα εικονοστοιχεία θορύβου. Επίσης παρατηρούμε ότι σε αυτό το παράδειγμα το δίκτυο με τελικό σφάλμα 0.1 παρουσιάζει αρκετά σφάλματα σε σχέση με τα προηγούμενα παραδείγματα.

5.4 Αποτελέσματα training set 2

Σε αυτό το training set αφήσαμε εικόνες με περισσότερα μαύρα εικονοστοιχεία για περισσότερη λεπτομέρεια. Ήταν μια προσπάθεια να αφαιρέσουμε εικόνες με occlusion και επιπλέον θόρυβο που παραγόταν από την διαδικασία μείωσης των διαστάσεων των εικόνων.

Πείραμα 1^ο:



Σχολιασμός:

Αρχικά πρέπει να αναφερθεί ότι η πραγματική-αρχική εικόνα προσώπου, ήταν αρκετά καλή και πολύ καλύτερη από όλες τις εικόνες του προηγούμενου training set.

Στην εικόνα με κάθετο occlusion αποκρύψαμε λεπτομέρεια από το κάτω μέρος του δεξιού περιγράμματος και απο το δεξί μάτι. Παρατηρούμε ότι το δίκτυο με τελικό σφάλμα 0.01 συμπληρώνει άψογα και τα δύο. Το δίκτυο με σφάλμα 0.1 συμπληρώνει και αυτό και τα δύο όμως αλλάζει όλα τα

εικονοστοιχεία που βρίσκονταν στο κέντρο της εικόνας και αλλάζουν όλες οι εσωτερικές λεπτομέρειες του προσώπου.

Στην εικόνα του διαγώνιου occlusion αφαιρέσαμε λεπτομέρεια από το αριστερό μάτι και από το δεξί κάτω μέρος του περιγράμματος. Το δίκτυο με σφάλμα 0.01 τα πήγε αρκετά καλά όπως μπορούμε να παρατηρήσουμε και από την συμπληρωμένη εικόνα. Συμπλήρωσε τέλεια όλο το ήδη αποκομμένο μάτι της πραγματικής εικόνας και επιπλέον πρόσθεσε όλα τα εικονοστοιχεία στο ελλιπών περίγραμμα. Μπορούμε να πούμε ότι αφαίρεσε όλα τα εικονοστοιχεία της μύτης κάτι που ήταν απαραίστο και ανεπιθύμητο. Το δίκτυο με σφάλμα 0.1 έφτιαξε μια χειρότερη εικόνα σε σχέση το δίκτυο με τελικό σφάλμα 0.01 αφού πρόσθεσε και αφαίρεσε εικονοστοιχεία σε μη ζητούμενα σημεία.

Τέλος, στην εικόνα με το οριζόντιο occlusion αφαιρέσαμε κομμάτι στην δεξιά μεριά του περιγράμματος και στην αριστερή πλευρά του στόματος. Το δίκτυο με σφάλμα 0.01 συμπλήρωσε το δεξί περίγραμμα ικανοποιητικά και σχημάτισε ένα τέλειο στόμα. Το μόνο αρνητικό είναι ότι αφαίρεσε το αριστερό περίγραμμα. Το δίκτυο με τελικό σφάλμα 0.1 συμπλήρωσε ικανοποιητικά και τα δύο όμως δημιούργησε ελάχιστο θόρυβο στην μέση του προσώπου και συγκεκριμένα στην μύτη και το στόμα.

Ως γενική παρατήρηση για τα πιο πάνω μπορούμε να πούμε ότι το δίκτυο με σφάλμα 0.01 τα πήγε αρκετά καλά.

Στατιστική Μελέτη Πειράματος 1:

	ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΚΑΘΕΤΟΥ OCCLUSION		ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΔΙΑΓΩΝΙΟΥ OCCLUSION		ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΟΡΙΖΟΝΤΙΟΥ OCCLUSION	
	Συνολικά	%	Συνολικά	%	Συνολικά	%
Τελικό Σφάλμα 0.1	111(56)	27.75	138(82)	34.5	137(77)	34.25
Τελικό Σφάλμα 0.01	73(30)	18.25	73(23)	18.25	56(14)	14

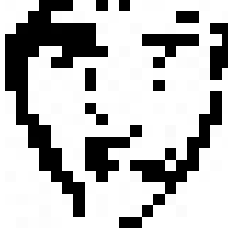
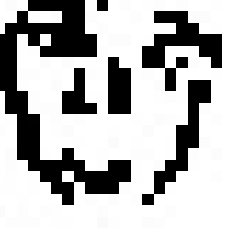
Πίνακας 5.4.1

*Στην παρένθεση είναι τα εικονοστοιχεία θορύβου (Μαύρα εικονοστοιχεία στην εικόνα εξόδου που δεν υπήρχαν στην πραγματική εικόνα).

Σχολιασμός στατιστικής Μελέτης Πειράματος 1:

Παρατηρούμε από τον πιο πάνω πίνακα ότι τόσο τα λανθασμένα εικονοστοιχεία στην εκτέλεση του δικτύου με τελικό σφάλμα 0.01 όσο και γενικά τα λανθασμένα εικονοστοιχεία μειώνονται σε σχέση με το δίκτυο με τελικό σφάλμα 0.1. Παρατηρούμε επίσης ότι στην περίπτωση δικτύου με τελικό σφάλμα 0.01, τα εικονοστοιχεία θορύβου μειώνονται και αυτά.

Πείραμα 2°:

ΠΡΑΓΜΑΤΙΚΗ ΕΙΚΟΝΑ	ΚΑΘΕΤΟ OCCLUSION	ΔΙΑΓΩΝΙΟ OCCLUSION	ΟΡΙΖΟΝΤΙΟ OCCLUSION
			
ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΙΚΤΥΟΥ RBM ΜΕ ΤΕΛΙΚΟ ΤΕΤΡΑΓΩΝΙΚΟ ΣΦΑΛΜΑ 0.1			
			

Σχολιασμός:

Αρχικά πρέπει να αναφερθεί ότι η πραγματική-αρχική εικόνα προσώπου, ήταν αρκετά καλή και πολύ καλύτερη από όλες τις εικόνες του προηγούμενου training set.

Στην εικόνα με κάθετο occlusion αποκρύψαμε λεπτομέρεια από το κάτω μέρος του δεξιού περιγράμματος και από το δεξί μάτι. Παρατηρούμε ότι το δίκτυο με τελικό σφάλμα 0.01 συμπληρώνει άψογα και τα δύο. Το δίκτυο με σφάλμα 0.1 συμπληρώνει και αυτό και τα δύο όμως αλλάζει όλα τα εικονοστοιχεία που βρίσκονταν στο κέντρο της εικόνας και αλλάζουν όλες οι εσωτερικές λεπτομέρειες του προσώπου.

Στην εικόνα του διαγώνιου occlusion αφαιρέσαμε λεπτομέρεια από το αριστερό μάτι και από το δεξί κάτω μέρος του περιγράμματος. Το δίκτυο με σφάλμα 0.01 τα πήγε αρκετά καλά όπως μπορούμε να παρατηρήσουμε και από την συμπληρωμένη εικόνα. Συμπλήρωσε τέλεια όλο το ήδη αποκομμένο μάτι της πραγματικής εικόνας και επιπλέον πρόσθεσε όλα τα εικονοστοιχεία στο ελλιπών περίγραμμα. Μπορούμε να πούμε ότι αφαίρεσε όλα τα εικονοστοιχεία της μύτης κάτι που ήταν αχρείαστο και ανεπιθύμητο. Το δίκτυο με σφάλμα 0.1 έφτιαξε μια χειρότερη εικόνα σε σχέση το δίκτυο με τελικό σφάλμα 0.01 αφού πρόσθεσε και αφαίρεσε εικονοστοιχεία σε μη ζητούμενα σημεία.

Τέλος, στην εικόνα με το οριζόντιο occlusion αφαιρέσαμε κομμάτι στην δεξιά μεριά του περιγράμματος και στην αριστερή πλευρά του στόματος. Το δίκτυο με σφάλμα 0.01 συμπλήρωσε το δεξί περίγραμμα ικανοποιητικά και σχημάτισε ένα τέλειο στόμα. Το μόνο αρνητικό είναι ότι αφαίρεσε το αριστερό περίγραμμα. Το δίκτυο με τελικό σφάλμα 0.1 συμπλήρωσε ικανοποιητικά και τα δύο όμως δημιούργησε ελάχιστο θόρυβο στην μέση του προσώπου και συγκεκριμένα στην μύτη και το στόμα.

Ως γενική παρατήρηση για τα πιο πάνω μπορούμε να πούμε ότι το δίκτυο με σφάλμα 0.01 τα πήγε αρκετά καλά.

Στατιστική Μελέτη Πειράματος 2:

	ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΚΑΘΕΤΟΥ OCCLUSION		ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΔΙΑΓΩΝΙΟΥ OCCLUSION		ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΟΡΙΖΟΝΤΙΟΥ OCCLUSION	
	Συνολικά	%	Συνολικά	%	Συνολικά	%
Τελικό Σφάλμα 0.1	101(51)	25.25	133(54)	33.25	141(66)	35.25
Τελικό Σφάλμα 0.01	90(30)	22.5	96(39)	24	94(32)	23.5

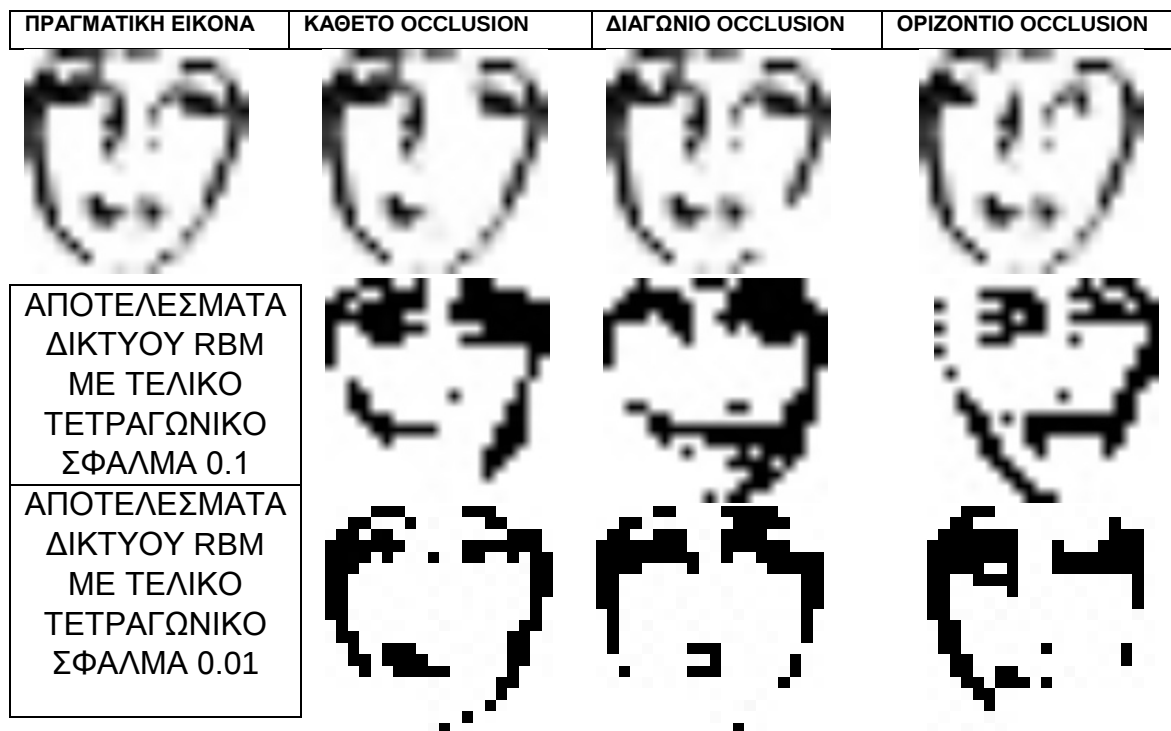
Πίνακας 5.4.2

***Στην παρένθεση είναι τα εικονοστοιχεία θορύβου (Μαύρα εικονοστοιχεία στην εικόνα εξόδου που δεν υπήρχαν στην πραγματική εικόνα).**

Σχολιασμός στατιστικής Μελέτης Πειράματος 2:

Παρατηρούμε από τον πιο πάνω πίνακα ότι τα λανθασμένα εικονοστοιχεία και ο θόρυβος στην εκτέλεση του δικτύου με τελικό σφάλμα 0.01 μειώνονται σε σχέση με το δίκτυο με τελικό σφάλμα 0.1 όμως όχι ικανοποιητικά.

Πείραμα 3°:



Σχολιασμός:

Αρχικά πρέπει να αναφερθεί ότι η πραγματική-αρχική εικόνα προσώπου, ήταν αρκετά καλή και πολύ καλύτερη από όλες τις εικόνες του προηγούμενου training set.

Στην εικόνα με κάθετο occlusion αποκρύψαμε λεπτομέρεια από την μύτη και το στόμα. Παρατηρούμε ότι το δίκτυο με τελικό σφάλμα 0.01 συμπληρώνει αρκετά καλά και τα δύο. Το δίκτυο με σφάλμα 0.1 συμπληρώνει ελαφρώς το στόμα με αρκετή δόση θορύβου. Είναι γενικά μια πολύ κακή εικόνα για το δίκτυο.

Στην εικόνα του διαγώνιου occlusion αφαιρέσαμε λεπτομέρεια από το αριστερό μάτι και από το δεξί κάτω μέρος του περιγράμματος. Το δίκτυο με σφάλμα 0.01 τα πήγε σχετικά καλά όπως μπορούμε να παρατηρήσουμε και από την συμπληρωμένη εικόνα. Συμπλήρωσε μερικά εικονοστοιχεία από την αποκρυμμένη λεπτομέρεια του ματιού και το αποκομμένο περίγραμμα με μερική δόση θορύβου. Το δίκτυο με σφάλμα 0.1 έφτιαξε μια χειρότερη εικόνα σε σχέση

το δίκτυο με τελικό σφάλμα 0.01 αφού πρόσθεσε εικονοστοιχεία στο κέντρο του προσώπου και το αποτέλεσμα δεν θυμίζει πρόσωπο.

Τέλος, στην εικόνα με το οριζόντιο occlusion αφαιρέσαμε λεπτομέρεια από τα δυο μάτια. Το δίκτυο με σφάλμα 0.01 συμπλήρωσε και τα δύο μάτια αρκετά καλά. Το δίκτυο με τελικό σφάλμα 0.1 συμπλήρωσε με αρκετό θόρυβο και τα δύο μάτια και πρόσθεσε αρκετό θόρυβο στο δεξί περίγραμμα.

Ως γενική παρατήρηση για τα πιο πάνω μπορούμε να πούμε ότι το δίκτυο με σφάλμα 0.01 τα πήγε αρκετά καλά.

Στατιστική Μελέτη Πειράματος 3:

	ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΚΑΘΕΤΟΥ OCCLUSION		ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΔΙΑΓΩΝΙΟΥ OCCLUSION		ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΟΡΙΖΟΝΤΙΟΥ OCCLUSION	
	Συνολικά	%	Συνολικά	%	Συνολικά	%
Τελικό Σφάλμα 0.1	147(84)	36.25	125(87)	31.25	104(66)	26
Τελικό Σφάλμα 0.01	69(38)	17.25	73(39)	18.25	72(41)	18

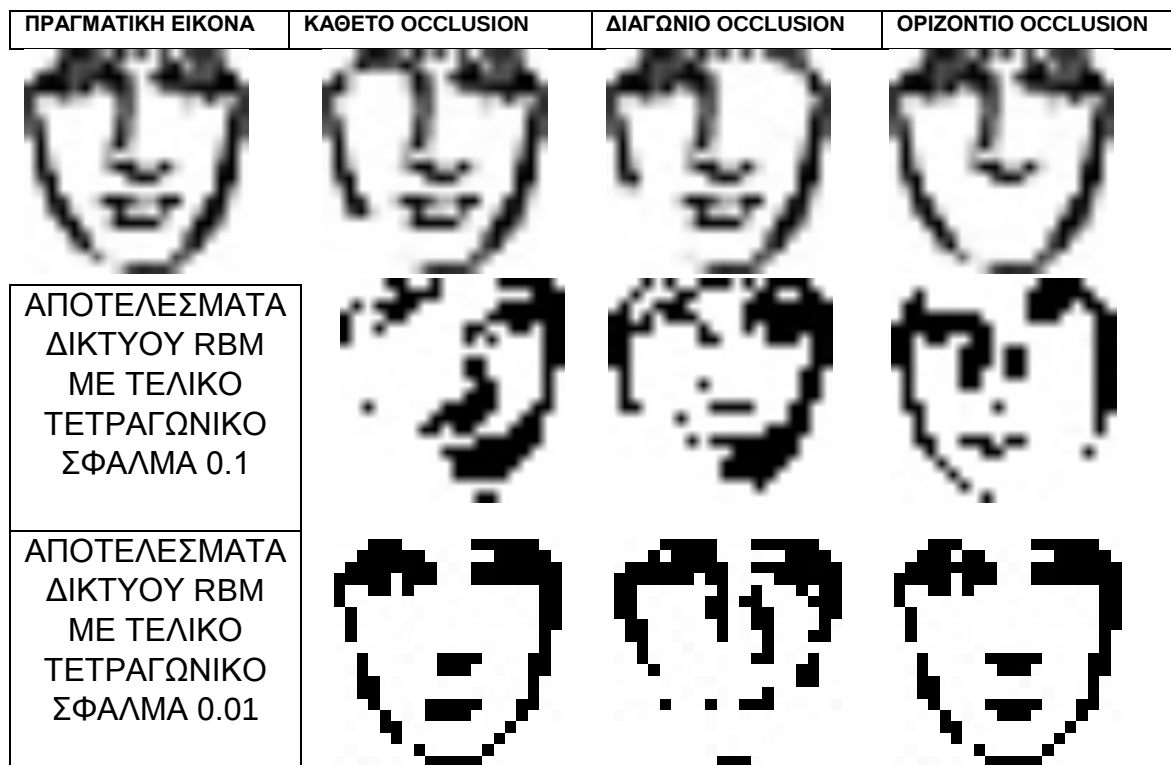
Πίνακας 5.4.3

*Στην παρένθεση είναι τα εικονοστοιχεία θορύβου (Μαύρα εικονοστοιχεία στην εικόνα εξόδου που δεν υπήρχαν στην πραγματική εικόνα).

Σχολιασμός στατιστικής Μελέτης Πειράματος 3:

Παρατηρούμε από τον πιο πάνω πίνακα ότι τα λανθασμένα εικονοστοιχεία και ο θόρυβος στην εκτέλεση του δικτύου με τελικό σφάλμα 0.01 μειώνονται αρκετά σε σχέση με το δίκτυο με τελικό σφάλμα 0.1.

Πείραμα 4^ο:



Σχολιασμός:

Αρχικά πρέπει να αναφερθεί ότι η πραγματική-αρχική εικόνα προσώπου, ήταν αρκετά καλή και πολύ καλύτερη από όλες τις εικόνες του προηγούμενου training set.

Επειδή σε αυτό το παράδειγμα όλες οι εικόνες των αποτελεσμάτων είναι απογοητευτικές ειδικά του δικτύου με τελικό σφάλμα 0.1 δεν θα ασχοληθούμε με την αναλυτική περιγραφή των εικόνων του δικτύου. Είναι ένα παράδειγμα που πρέπει να συμβαίνει πάρα πολύ σπάνια. Οι εικόνες που συμπληρώνονται σχετικά καλά είναι οι εικόνες του δικτύου με τελικό σφάλμα 0.01. Συμπληρώνονται οι αποκρυμμένες λεπτομέρειες με μικρή δόση θορύβου όπως φαίνεται και στον πίνακα 5.4.4.

Στατιστική Μελέτη Πειράματος 4:

	ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΚΑΘΕΤΟΥ OCCLUSION		ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΔΙΑΓΩΝΙΟΥ OCCLUSION		ΤΕΛΙΚΗ ΕΙΚΟΝΑ ΟΡΙΖΟΝΤΙΟΥ OCCLUSION	
	Συνολικά	%	Συνολικά	%	Συνολικά	%
Τελικό Σφάλμα 0.1	110(41)	27,5	99(44)	24.75	109(44)	24.75
Τελικό Σφάλμα 0.01	57(18)	14.25	93(33)	23.25	56(18)	14

Πίνακας 5.4.4

***Στην παρένθεση είναι τα εικονοστοιχεία θορύβου (Μαύρα εικονοστοιχεία στην εικόνα εξόδου που δεν υπήρχαν στην πραγματική εικόνα).**

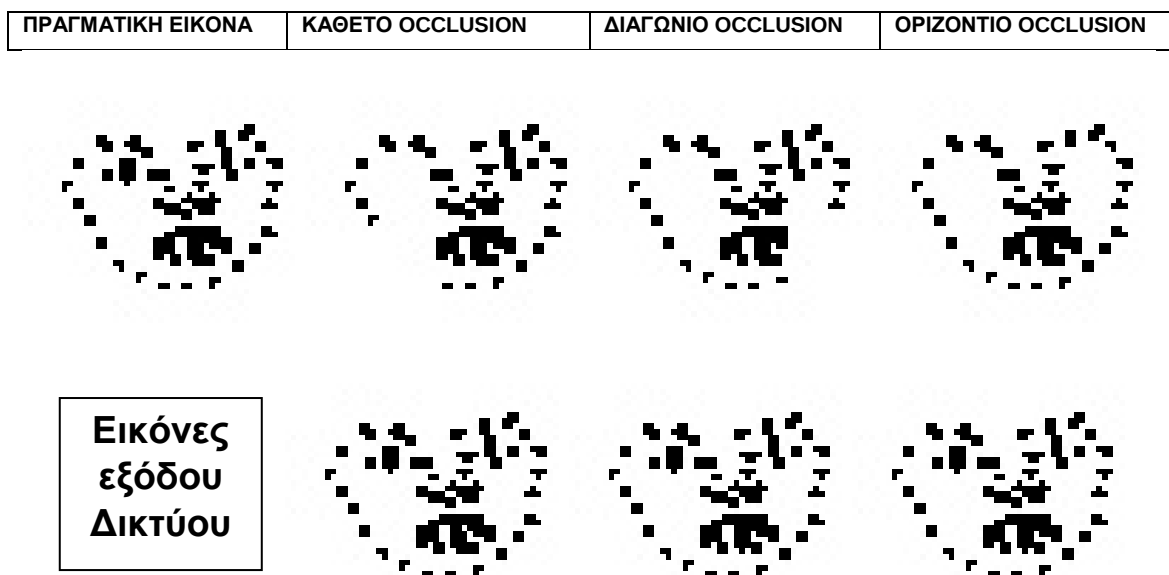
Σχολιασμός στατιστικής Μελέτης Πειράματος 4:

Παρατηρούμε από τον πιο πάνω πίνακα ότι τα λανθασμένα εικονοστοιχεία στην εκτέλεση του δικτύου με τελικό σφάλμα 0.01 μειώνονται δραματικά σε σχέση με το δίκτυο με τελικό σφάλμα 0.1. Παρατηρούμε ότι σε αυτό το πείραμα μειώνεται δραματικά και ο θόρυβος. Αξίζει να σημειωθεί ότι τα λανθασμένα εικονοστοιχεία των εικόνων του δικτύου με τελικό σφάλμα 0.1 είναι σε αρκετά υψηλά επίπεδα.

5.5 Αποτελέσματα με τις εικόνες σημείων

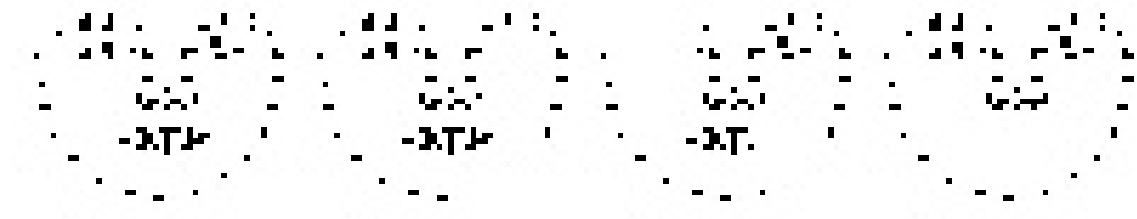
Μετά από αρκετά πειράματα με εικόνες προσώπων με λεπτομέρειες διαπιστώσαμε ότι το δίκτυο μας συμπλήρωνε τις εικόνες με αρκετό θόρυβο. Αυτός ο θόρυβος προερχόταν από την χαμηλή ποιότητα των εικόνων. Με αυτό εννοούμε ότι στην προσπάθειά μας να μειώσουμε τις διαστάσεις των εικόνων, τα μαύρα εικονοστοιχεία, που ήταν η λεπτομέρεια στις εικόνες, πλησίαζαν μεταξύ τους με αποτέλεσμα να έχουμε μεγάλη απώλεια πληροφορίας με την συνένωση των μαύρων εικονοστοιχείων και ως αποτέλεσμα να γίνεται το αποτέλεσμα λιγότερο ποιοτικό. Έτσι αποφασίσαμε να δοκιμάσουμε το δίκτυο με εικόνες σημείων. Μια μικρή ομάδα από σημεία σχηματίζει μια λεπτομέρεια στο πρόσωπο. Πιο κάτω παρουσιάζουμε τα αποτελέσματα που λάβαμε από τρία πειράματα που κάναμε.

Πείραμα 1^ο:

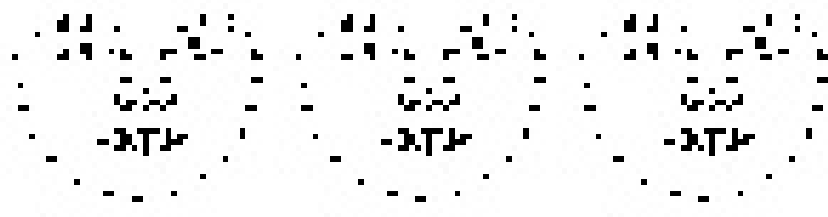


Πείραμα 2°:

ΠΡΑΓΜΑΤΙΚΗ ΕΙΚΟΝΑ	ΚΑΘΕΤΟ OCCLUSION	ΔΙΑΓΩΝΙΟ OCCLUSION	ΟΡΙΖΟΝΤΙΟ OCCLUSION
-------------------	------------------	--------------------	---------------------

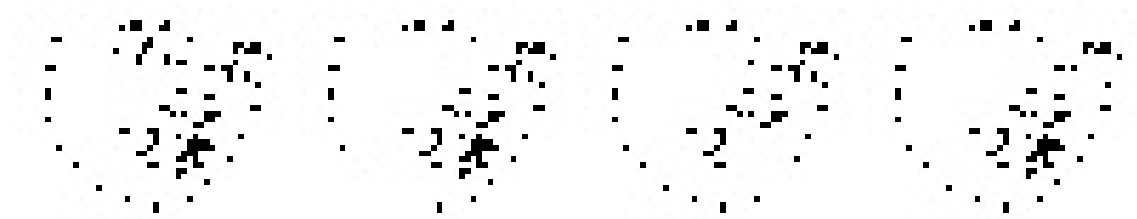


Εικόνες
εξόδου
Δικτύου

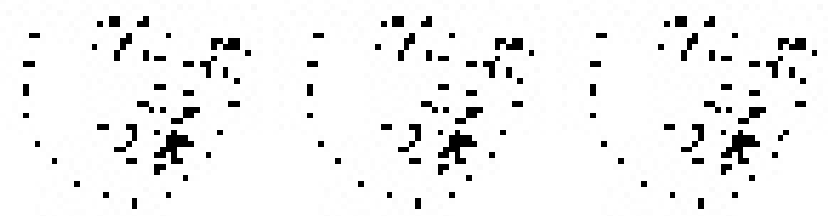


Πείραμα 3°:

ΠΡΑΓΜΑΤΙΚΗ ΕΙΚΟΝΑ	ΚΑΘΕΤΟ OCCLUSION	ΔΙΑΓΩΝΙΟ OCCLUSION	ΟΡΙΖΟΝΤΙΟ OCCLUSION
-------------------	------------------	--------------------	---------------------



Εικόνες
εξόδου
Δικτύου



Στα πιο πάνω πειράματα αποκόψαμε πληροφορία κάθετα, διαγώνια και οριζόντια για την κάθε ξεχωριστή εικόνα του κάθε πειράματος. Παρατηρούμε ότι το δίκτυο στις εικόνες με τα σημεία τα βρίσκει αρκετά εύκολα αφού συμπληρώνει όλες τις εικόνες σε σημείο που έχουν ελάχιστες διαφορές από τις πραγματικές εικόνες. Για τα πιο πάνω πειράματα δοκιμάσαμε να μετρήσουμε την διαφορά από τις πραγματικές εικόνες και τα αποτελέσματα φαίνονται στον πίνακα 5.5.1. Ο τρόπος μέτρησης της διαφοράς αυτής είναι ο εξής. Εάν δύο εικονοστοιχεία στις δύο αντίστοιχες εικόνες(εξόδου, πραγματικής) έχουν διαφορετικό χρώμα, τότε η διαφορά αυξάνεται κατά ένα.

	Διαφορά πραγματικής-κάθετου occlusion		Διαφορά πραγματικής- διαγώνιου occlusion		Διαφορά πραγματικής- οριζόντιου occlusion	
	Ανά εικόνα	%	Ανά εικόνα	%	Ανά εικόνα	%
Πείραμα 1°	1	0.03	1	0.03	1	0.03
Πείραμα 2°	1	0.03	1	0.03	2	0.06
Πείραμα 3°	0	0	0	0	1	0.03

Πίνακας 5.5.1 Μπορούμε πλέον και στατιστικά να πούμε ότι το δίκτυο μας ήταν σχεδόν αλάνθαστο στην εκτέλεση με τα σημεία προσώπου. Η μέγιστη διαφορά μεταξύ πραγματικής εικόνας και εικόνας εξόδου από το δίκτυο ήταν μόνο 2 εικονοστοιχεία και ποσοστό 0.06. Αξίζει να σημειώσουμε ότι σε δύο περιπτώσεις το δίκτυο ήταν αλάνθαστο.

Παρατηρούμε ότι οι τελικές εικόνες είναι πανομοιότυπες με τις πραγματικές με ελάχιστες διαφορές μεταξύ τους. Βλέπουμε ότι το δίκτυο κατανόησε πλήρως την θέση των σημείων στον χώρο και συμπλήρωνε το οποιοδήποτε ελλείπων σημείο.

Κεφάλαιο 6

Συμπεράσματα

Γενικά, μπορούμε να πούμε ότι όσο αφορά τα δεδομένα εικόνων με λεπτομέρειες το δίκτυο RBM φάνηκε να αναγνωρίζει το occlusion στις εικόνες σε όλες τις εικόνες δοκιμής, και να προσπαθεί να τις συμπληρώσει. Αρκετές φορές η συμπληρωμένη εικόνα ήταν αρκετά καλή και σε μερικές περιπτώσεις καλύτερη και από την πραγματική εικόνα. Όλα τα αποτελέσματα είχαν ένα σχετικά υψηλό ποσοστό θορύβου και αυτό ήταν αποτέλεσμα της συμπίεσης των εικόνων που έγινε κατά την φάση της προ επεξεργασίας δεδομένων. Επιπρόσθετα στα αποτελέσματα του training set 1 ο θόρυβος πιθανό να προέρχεται και από την ανομοιογένεια των εικόνων στα δεδομένα εκπαίδευσης. Αρκετή πληροφορία αφαιρέθηκε κατά την διαδικασία αυτή κάτι που είχε σαν αποτέλεσμα οι εικόνες των προσώπων να ήταν αρκετά χαμηλής ποιότητας. Θεωρούμε ότι στην βέλτιστη περίπτωση που οι εικόνες δεν συμπιέζονταν τα αποτελέσματα θα ήταν αρκετά καλύτερα και πολύ πιο ρεαλιστικά. Το πιστεύουμε αυτό, λόγω των αποτελεσμάτων των εικόνων με τα σημεία. Σε αυτές τις εικόνες το δίκτυο μας έδειξε να κατανοεί πλήρως την γεωμετρία πίσω από τα σημεία και συμπλήρωσε σχεδόν αλάνθαστα όλα τα σημεία που απουσίαζαν. Γενικά το δίκτυο RBM είναι ένα νευρωνικό δίκτυο όπου μπορεί εύκολα να χρησιμοποιηθεί σαν παραγωγικό μοντέλο και είναι κάτι που αποδεικνύεται και από τα αποτελέσματα μας. Υπό άλλες συνθήκες και με την διαθεσιμότητα περισσότερης υπολογιστικής ισχύος θα μπορούσε το δίκτυο μας να δοκιμαστεί και σε δεδομένα υψηλότερης ποιότητας. Σε αυτή την περίπτωση θα μπορούσαμε με βεβαιότητα να κρίνουμε κατά πόσον τα δίκτυα RBM μπορούν βέλτιστα να χρησιμοποιηθούν σε αυτό τον τομέα με εικόνες με αρκετή λεπτομέρεια. Σε εικόνες με λίγη λεπτομέρεια όπως αυτές των εικόνων με σημεία το RBM έδειξε να αντεπεξέρχεται πλήρως.

Κεφάλαιο 7

Μελλοντική Εργασία

Όσο αφορά την μελλοντική εργασία που μπορεί να γίνει είναι η επέκταση του μοντέλου με την χρήση ενός ακόμη νευρωνικού δικτύου, ίσως ενός Convolutional Neural network, το οποίο θα μας επιτρέψει να εισάγουμε έγχρωμες εικόνες έτσι τα αποτελέσματα μας θα έχουν έγχρωμη μορφή.

Αναλυτικά μπορεί το υπάρχον σύστημα μας να υπολογίζει τα σημεία κλειδιά στην εικόνα προσώπου και το convolutional δίκτυο θα επεξεργάζεται πιο σύνθετες πληροφορίες όπως αυτές του χρώματος των διάφορων χαρακτηριστικών του προσώπου. Έτσι με κάποιο τρόπο το Convolutional δίκτυο θα παίρνει την έξοδο από το Boltzmann δίκτυο και αφού πλέον θα είναι το πρόβλημα του occlusion λυμένο, το convolutional θα χρωματίζει και θα προσθέτει τις λεπτομέρειες που χρειάζεται στην κάθε εικόνα. Κάπου εδώ θα πρέπει να αναφέρουμε ότι εάν ο πιο πάνω συνδυασμός πετύχει τότε το δίκτυο θα μπορεί να χρησιμοποιηθεί και στον τομέα της γήρανσης του ανθρώπινου προσώπου (Face aging).

Ένας άλλος τρόπος για να πετύχει το πιο πάνω είναι η τροποποίηση του υπάρχοντος συστήματος έτσι ώστε να μπορεί να λειτουργεί με πραγματικές τιμές δεδομένων και όχι δυαδικές. Όταν αναφερόμαστε σε πραγματικές τιμές δεδομένων εννοούμε έγχρωμες εικόνες εισόδου. Αναλυτικά σε αυτή την περίπτωση τα δεδομένα θα κωδικοποιούνται με κανονικοποίηση στο εύρος των πραγματικών τιμών στο διάστημα 0,1. Δηλαδή τα δεδομένα μας στο κάθε vector εισόδου θα τριπλασιαστούν και αντί φωτεινότητα μαύρου θα έχουμε την κανονική μορφή φωτεινότητας RGB όπου το κάθε εικονοστοιχείο θα καταλαμβάνει 3 συνεχόμενες θέσεις. Στην πρώτη θέση θα είναι η κωδικοποιημένη τιμή της φωτεινότητας του κόκκινου, η δεύτερη του πράσινου και η τρίτη του μπλε. Στην περίπτωση RBM με πραγματικές τιμές δεδομένων

με ένα μικρό «κόλπο» εκπαιδεύεται το δίκτυο με πραγματικές τιμές. Λεπτομέρειες για την χρήση RBM με πραγματικές τιμές δεδομένων, περιγράφεται στο άρθρο [16] στην ενότητα 6.

Παρομοίως με το πιο πάνω, για αποτελέσματα με έγχρωμες εικόνες θα μπορούσε να χρησιμοποιηθεί ένα Gaussian Restricted Boltzmann machine αντί του RBM. Το RGBM έχει την ιδιαιτερότητα να δουλεύει με πραγματικές τιμές δεδομένων όπως περιγράφεται στο άρθρο [16] στην ενότητα 6, αλλά και στο άρθρο [20] στην παράγραφο Continuous RBMs.

ΑΝΑΦΟΡΕΣ

[1] LeCun Y., Bengio Y. and Hinton G. (2015) Deep learning. *Nature*, 521, 436-444.

[2] Ekenel, H. K. and Stiefelhagen, R. (2009). Why Is Facial Occlusion a Challenging Problem? In: Tistarelli, M., Nixon, M.S. (eds), *ICB 2009: Advances in Biometrics. Lecture Notes in Computer Science*, vol 5558. pp. 299-308, Springer, Berlin, Heidelberg.

[3] Hinton, G. E. (2010). A Practical Guide to Training Restricted Boltzmann Machines, Technical Report, UTML TR 2010–003, Department of Computer Science, University of Toronto.

[4] Sun, Y., Wang, X. and Tang, X. (2013). Deep Convolutional Network Cascade for Facial Point Detection. *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, USA, pp. 3476-3483. doi: 10.1109/CVPR.2013.446

[5] Zhu, Z., Luo, P., Wang, X. and Tang, X. (2014). Recover Canonical-View Faces in the Tild with Deep Neural Networks. *arXiv preprint*, arXiv:1404.3543

[6] Tang, Y., Salakhutdinov, R. and Hinton, G. (2012). Robust Boltzmann Machines for Recognition and Denoising. *Proceedings of the 2012 IEEE conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, pp. 2264-2271. doi: 10.1109/CVPR.2012.6247936.

[7] Wu, Y., Wang, Z. and Ji, Q. (2013). Facial Feature Tracking under Varying Facial Expressions and Face Poses based on Restricted Boltzmann Machines. *Proceedings of the 2013 IEEE conference on Computer Vision and Pattern Recognition*, Portland, OR, USA, pp. 3452-3459. doi: 10.1109/CVPR.2013.443.

- [8] Shu, X., Tang, J., Lai, H., Liu, L. and Yan, S. (2015). Personalized age progression with aging dictionary, arXiv preprint arXiv: arXiv:1510.06503v1
- [9] CASD Database <https://www.casd.eu/en/available-data-on-casd/>
- [10] MORPH Database <https://www.faceaginggroup.com/morph/>
- [11] FGNET Database https://fipa.cs.kit.edu/433_451.php
- [12] Antipov, G., Baccouche, M. and Dugelay, J-L. (2017). Face aging with conditional generative adversarial networks, arXiv preprint arXiv: 1702.01983v2
- [13] Zhang, Z., Song, Y. and Qi, H. (2017). Age Progression/Regression by Conditional Adversarial Autoencoder, arXiv preprint arXiv: arXiv:1702.08423v2
- [14] https://www.researchgate.net/profile/Andreas_Lanitis/2
- [15] Hinton, G. (2001). Training Products of Experts by Minimizing Contrastive Divergence. Neural Computation, 14(8), 1771-1800
- [16] Fisher, A. and Igel, C. (2014). Training restricted Boltzmann Machines: An introduction. Pattern Recognition, 47(1), 25-39.
- [17] Hinton, G. E. (2007). Boltzmann Machines, Scholarpedia, 2(5): 1668.
- [14] https://www.dropbox.com/s/ja9fadc48gvxo95/IMAGES_PERSONS_001-012.zip?dl=0&file_subpath=%2FIMAGES_PERSONS_001-012
- [19] <http://yann.lecun.com/exdb/mnist/>
- [20] <https://deeplearning4j.org/restrictedboltzmannmachine>

Παράρτημα – Κώδικας

```
from __future__ import print_function
from multiprocessing import Process
from VectoriseImages import load_data
import numpy as np # Using for vector processing.
import cv2 # Using this library to read images and to convert the output of
sampling as images.
import time

class RBM:
    def __init__(self, num_visible, num_hidden):
        self.num_hidden = num_hidden
        self.num_visible = num_visible
        self.debug_print = True

        '''
        Initialize a weight matrix, of dimensions (num_visible x num_hidden),
        using a uniform distribution between -sqrt
        (6. / (num_hidden + num_visible)) and sqrt(6. / (num_hidden +
        num_visible)). One could vary the standard
        deviation by multiplying the interval with appropriate value. Here we
        initialize the weights with mean 0 and
        standard deviation 0.1.
        '''
        np_rng = np.random.RandomState (1234)

        self.weights = np.asarray (np_rng.uniform (low=-0.1 * np.sqrt (6. /
        (num_hidden + num_visible)),
                                                    high=0.1 * np.sqrt (6. /
        (num_hidden + num_visible)),
                                                    size=(num_visible,
        num_hidden)))

        # Insert weights for the bias units into the first row and first
        column.
        self.weights = np.insert (self.weights, 0, 0, axis=0)
        self.weights = np.insert (self.weights, 0, 0, axis=1)

    # def loadW(self, file):
    #     fp = open (file, "r")
    #     lst=[]
    #     lst.append((fp.readlines(1000000)))
    #     print (lst)

    def train(self, finalError, data, learning_rate):
        """na
            Train the machine.

            Parameters
            -----

            datavector: A matrix where each row is a training
            example consisting of the states of visible units.

            max_epochs: The total epochs of training

            learning_rate: The learning rate

            num_examples: The total training data
```

```

"""

num_examples = len (data)
print (num_examples)
# Insert bias units of 1 into the first column.

data = np.insert (data, 0, 1, axis=1)
error = 1
epoch = 1
while (error > finalError):
    # for epoch in range (max_epochs):

        # Clamp to the data and sample from the hidden units.

        # (This is the "positive CD phase", aka the reality phase.)

        pos_hidden_activations = np.dot (data, self.weights)

        pos_hidden_probs = self._logistic (pos_hidden_activations)

        #pos_hidden_probs[:, 0] = 1 # Fix the bias unit.

        pos_hidden_states = pos_hidden_probs > np.random.rand
(num_examples, self.num_hidden + 1)

        # Note that we're using the activation *probabilities* of the
hidden states, not the hidden states

        # themselves, when computing associations. We could also use the
states; see section 3 of Hinton's

        # "A Practical Guide to Training Restricted Boltzmann Machines"
for more.

        pos_associations = np.dot (data.T, pos_hidden_probs)

        # Reconstruct the visible units and sample again from the hidden
units.

        # (This is the "negative CD phase", aka the daydreaming phase.)

        neg_visible_activations = np.dot (pos_hidden_states,
self.weights.T)

        neg_visible_probs = self._logistic (neg_visible_activations)

        #neg_visible_probs[:, 0] = 1 # Fix the bias unit.

        neg_hidden_activations = np.dot (neg_visible_probs, self.weights)

        neg_hidden_probs = self._logistic (neg_hidden_activations)

        # Note, again, that we're using the activation *probabilities*
when computing associations, not the states

        # themselves.

        neg_associations = np.dot (neg_visible_probs.T, neg_hidden_probs)

        # Update weights.

        self.weights += learning_rate * ((pos_associations -
neg_associations) / num_examples)
        # initial_lrate = 0.1
        # drop = 0.5
        # epochs_drop = 100000.0

```

```

        # learning_rate = initial_lrate ** (epoch/epochs_drop)

        error = np.sum ((data - neg_visible_probs) ** 2) / num_examples
        if self.debug_print:
            print ("Epoch %s: error is %s" % (epoch, error))
            if 0 == (epoch % 100):
                if epoch == 100:
                    errorFp = open ("error.txt", "w")
                else:
                    errorFp = open ("error.txt", "a")
                errorFp.write ("%f\n" % error)

                errorFp.close()

        epoch += 1

        weightsFp = open ("weights.txt", "w")
        weightsFp.writelines (["%s\n" % item for item in self.weights])
        weightsFp.close()

    def sampleRBM(self, data, steps):
        """
            Train the machine.

            Parameters
            -----

            data: A matrix where each row is a training example consisting
of the states of visible units.

            steps: The total epochs of training

            """
        num_examples = len (data)
        pos_hidden_states = np.ones (num_examples)

        for i in range (1, steps): # GIBBS SAMPLING
            if i == 1:
                data = np.insert (data, 0, 1)
                hidden_activations = np.dot (data, self.weights)
            else:
                hidden_activations = np.dot (neg_visible_activations,
self.weights)

            pos_hidden_probs = self._logistic (hidden_activations)
            pos_hidden_states = pos_hidden_probs > np.random.rand
(num_examples, self.num_hidden + 1)
            neg_visible_activations = np.dot (pos_hidden_states,
self.weights.T)
            neg_visible_probs = self._logistic (neg_visible_activations)
            neg_visible_probs[0] = 1 # Fix the bias unit.
            neg_hidden_activations = np.dot (neg_visible_probs, self.weights)
            neg_hidden_probs = self._logistic (neg_hidden_activations)

        return pos_hidden_probs

    def _logistic(self, x):
        return 1.0 / (1.0 + np.exp (-x))

    def run_visible(self, data):
        """
            Assuming the RBM has been trained (so that weights for the network
have been learned),

```

```

        run the network on a set of visible units, to get a sample of the
        hidden units.

    Parameters
    -----
    data: A matrix where each row consists of the states of the visible
    units.

    Returns
    -----
    hidden_states: A matrix where each row consists of the hidden units
    activated from the visible
    units in the data matrix passed in.
    """
    num_examples = len (data)

    # Create a matrix, where each row is to be the hidden units (plus a
    bias unit)

    # sampled from a training example.
    hidden_states = np.ones ((num_examples, self.num_hidden + 1))
    # Insert bias units of 1 into the first column of data.
    data = np.insert (data, 0, 1, axis=1)
    # Calculate the activations of the hidden units.
    hidden_activations = np.dot (data, self.weights)
    # Calculate the probabilities of turning the hidden units on.
    hidden_probs = self._logistic (hidden_activations)
    # Turn the hidden units on with their specified probabilities.
    hidden_states[:, :] = hidden_probs > np.random.rand (num_examples,
self.num_hidden + 1)
    # Always fix the bias unit to 1.
    # hidden_states[:,0] = 1

    # Ignore the bias units.
    hidden_states = hidden_states[:, 1:]
    return hidden_states

def run_hidden(self, data):
    """
    Assuming the RBM has been trained (so that weights for the network

```

```

have been learned),

    run the network on a set of hidden units, to get a sample of the
visible units.

Parameters
-----
data: A matrix where each row consists of the states of the hidden
units.

Returns
-----
visible_states: A matrix where each row consists of the visible units
activated from the hidden
units in the data matrix passed in.
"""

num_examples = len (data)

# Create a matrix, where each row is to be the visible units (plus a
bias unit)

# sampled from a training example.
visible_states = np.ones ((num_examples, self.num_visible + 1))

# Insert bias units of 1 into the first column of data.
data = np.insert (data, 0, 1, axis=1)

# Calculate the activations of the visible units.
visible_activations = np.dot (data, self.weights.T)

# Calculate the probabilities of turning the visible units on.
visible_probs = self._logistic (visible_activations)

# Turn the visible units on with their specified probabilities.
visible_states[:, :] = visible_probs > np.random.rand (num_examples,
self.num_visible + 1)

# Always fix the bias unit to 1.
# visible_states[:,0] = 1

# Ignore the bias units.
visible_states = visible_states[:, 1:]

return visible_states

def SamplingRBM(self, data, size, steps):
import os

```

```

        output_dir = 'C:\Users\Panayiotis\Dropbox\Computer
Science\PythonWorkspace\My RBM Implementation -- Dots\Output'
        if not os.path.exists (output_dir):
            os.mkdir (output_dir)
        os.chdir (output_dir)
        for i in range (1, steps):
            if i == 1:
                hs = r.run_visible (data)
            else:
                hs = r.run_visible (vs)

            vs = r.run_hidden (hs)

        for p in range (0, len (data)):
            fromVectorToImage (vs[p], p)

def fromVectorToImage(vector, p):
    img = np.ones (len (vector))
    # Insert data back to real form
    print (vector)
    newImg = np.ones (shape=(size, size))
    k = 0
    for i in range (0, size):
        for j in range (0, size):
            if vector[k] == 0:
                newImg[i, j] = 255
            else:
                newImg[i, j] = 0
            k = k + 1
    cv2.imwrite ('img' + str (p) + '.jpg', newImg)
    print ("img saved")

def normaliseInput(data, th):
    lengthData = len (data)
    lengthImg = len (data[0])
    for i in range (0, lengthData):
        for j in range (0, lengthImg):
            if data[i][j] > th:
                data[i][j] = 1
            else:
                data[i][j] = 0
    return data

if __name__ == '__main__':
    pathtrain = 'C:\Users\Panayiotis\Dropbox\Computer
Science\PythonWorkspace\My RBM Implementation -- Dots\Points\Training'
    pathtest = 'C:\Users\Panayiotis\Dropbox\Computer
Science\PythonWorkspace\My RBM Implementation -- Dots\Points\Testing'
    # Construction Of The RBM Network:
    size = 50
    gibbsSamplingSteps = 1000*10
    print ('Loading Data..')
    # Load The Images Data From My Function In File VectoriseImages.py:
    t0 = time.time ()
    training_data = [[0 for i in xrange (size ** 2)] for i in xrange (500)]
    testing_data = [[0 for i in xrange (size ** 2)] for i in xrange (9)]
    p1 = Process (target=load_data (size, pathtrain, 0, training_data))
    p1.start ()
    p1.join ()
    p2 = Process (target=load_data (size, pathtest, 1, testing_data))
    p2.start ()
    p2.join ()
    t1 = time.time ()
    print ("Time Programme consume on reading Data: ", t1 - t0)

```

```

TrainingDataSize = len (training_data)
TestingDataSize = len (testing_data)
print ('Training RBM with', TrainingDataSize, 'images')
print ('Testing RBM with', TestingDataSize, 'images')
print ('Constructing the Network')
r = RBM (num_visible=size ** 2, num_hidden=size ** 2)
fileW = 'C:\Users\Panayiotis\Dropbox\Computer Science\PythonWorkspace\My
RBM Implementation -- BINARY_FACES\weights.txt'
print ('All Done -- Ready for Training Phase Of Program')
t4 = time.time ()
r.train (0.01, training_data, 0.01)
t5 = time.time ()
print ("Time Programme consume on Training RBM: ", t5 - t4)
print ('Sampling Testing Data!!')
t6 = time.time ()
r.SamplingRBM (testing_data, size, gibbsSamplingSteps)
t7 = time.time ()
print ("Time Programme consume on Sampling Test Data: ", t7 - t6)
print ('Done!!')

def load_data(size, mypath, w, data):
    onlyfiles = [f for f in listdir (mypath) if isfile (join (mypath, f))]
    sorted (onlyfiles)
    images = np.empty (len (onlyfiles), dtype=object)
    a = 0
    for n in range (0, len (onlyfiles)):
        img = cv2.imread (join (mypath, onlyfiles[n]), 0)
        rows, cols = img.shape
        new_img = [0] * size * size
        k = 0
        for i in range (rows):
            for j in range (cols):
                if img[i, j]>128:
                    new_img[k] = 0
                else:
                    new_img[k] = 1
                k = k + 1
        data[a] = new_img
        a = a + 1

```

How to run the code:

- In folder "code" there are 3 different subfolders (Different uses of the implementation – Dot Images, binary images, Mnist Dataset)
- Subfolders has the same python files. The difference is the input of each implementation
- rbm.py is the file that contains the main class of each implementation
- every parameter can be found in main class of rbm.py as variable(gibbs sampling steps, size of images, final error)

- The only change that must be made is the change of pathtrain and pathtest variables(copy the link of your output folder and paste it to those variables)
- Also in function samplingRBM set the variable output_dir to your own folder.
- install any missing library