Πανεπιστήμιο Θεσσαλίας
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Οπτική Αναγνώριση Σιωπηρής Ομιλίας

Cued Speech Recognition

Διπλωματική Εργασία

ΔΡΑΚΟΥ ΣΤΑΡΟΥΛΑΣ

Επιβλέπων
Ποταμιάνος Γεράσιμος
Αναπληρωτής Καθηγητής

Βόλος, Ιούλιος 2019

Πανεπιστήμιο Θεσσαλίας
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Οπτική Αναγνώριση Σιωπηρής Ομιλίας

Cued Speech Recognition

Διπλωματική Εργασία

ΔΡΑΚΟΥ ΣΤΑΡΟΥΛΑΣ

Επιτροπή επίβλεψης

Επιβλέπων	Συνεπιβλέπων	Συνεπιβλέπων
Ποταμιάνος Γεράσιμος	Μπέλλας Νικόλαος	Βασιλακόπουλος Μιχαήλ
Αναπληρωτής Καθηγητής	Αναπληρωτής Καθηγητής	Αναπληρωτής Καθηγητής

Βόλος, Ιούλιος 2019

Πανεπιστήμιο Θεσσαλίας
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Ο συγγραφέας αυτής της εργασίας βεβαιώνει ότι κάθε βοήθεια την οποία είχε για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης βεβαιώνει ότι έχει αναφέρει τις όποιες πηγές από τις οποίες έκανε χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται επακριβώς, είτε παραφρασμένες.

Περίληψη

Τα τελευταία χρόνια έχει αναπτυχθεί ιδιαίτερα ο τομέας της όρασης υπολογιστών σε συνδυασμό με την χρήση νευρωνικών δικτύων, ώστε να επιλύσει διάφορα ζητήματα που αφορούν την επικοινωνία ανθρώπου-υπολογιστή. Μέσα από την μηχανική μάθηση, οι επιστήμονες έχουν καταφέρει να δημιουργήσουν μηχανές με ικανότητες αντίληψης του περιβάλλοντος ανάλογες με αυτές των ανθρώπων. Η παρούσα διπλωματική εργασία επεξεργάζεται δεδομένα από εικόνες βίντεο που σχετίζονται με την ανάλυση της σιωπηρής ομιλίας (Cued Speech) για την εξαγωγή συμπερασμάτων ως προς την προφορά φωνημάτων. Συγκεκριμένα στοχεύει με την βοήθεια εργαλείων όπως το OpenCV να εντοπίσει τις περιοχές ενδιαφέροντος (χέρια, χείλη, θέση χεριών) και με την χρήση συνελικτικών και απλών νευρωνικών δικτύων να εκπαιδεύσει 3 μοντέλα, που σε συνδυασμό θα προβλέπουν ποίο φώνημα προφέρεται. Η πραγματοποίηση της παραπάνω υλοποίησης και η επεξεργασία δεδομένων έχει δημιουργηθεί σε προγραμματιστικό περιβάλλον Python. Στην εργασία παρουσιάζεται αρχικά ένα εισαγωγικό σημείωμα για μια πρώτη γνωριμία με την σιωπηρή ομιλία και το θεωρητικό υπόβαθρο που απαιτείται για την κατανόηση της έννοιας των νευρωνικών δικτύων. Στη συνέχεια, παραδίδεται ο τρόπος προσέγγισης του προβλήματος, με αναλυτική περιγραφή όλων των απαραίτητων βημάτων για την προ-επεξεργασία των δεδομένων και αναλύεται εκτενώς η αρχιτεκτονική των δικτύων που χρησιμοποιήθηκαν. Τέλος, παρατίθενται τα αποτελέσματα από την υλοποίηση και η ανάλυσή τους.

Λέξεις Κλειδιά

Σιωπηρή Ομιλία, Python, OpenCV, Συνελικτικά Νευρωνικά Δίκτυα

Abstract

In recent years, the field of computer vision has been developed in conjunction with the use of neural networks to address various human-computer interaction problems. Through machine learning, scientists have managed to create computers with environmental perception similar to humans. This diploma thesis processes data from video images related to Cued Speech to recognize phonemes. Therefore, it tries to use tools - such as OpenCV - to identify areas of interest (hands, lips, positions of hands) and using convolutional and simple neural networks to train 3 models, which together predict which phoneme is pronounced. The procedure of the above implementation and data processing has been created in a Python programming environment. More specifically, first this diploma thesis presents an introductory note for a quick acquaintance with cued speech and the theoretical background needed to understand the concept of neural networks. Next, we provide a way to approach the problem with a detailed description of all the necessary steps for pre-processing the data and analyze extensively the architecture of the networks used. Finally, experimental results and their analysis are presented.

Keywords

Cued Speech, Python, OpenCV, Convolutional Neural Networks

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω θερμά τον καθηγητή κ. Γεράσιμο Ποταμιάνο για τις καίριες συμβουλές και την καθοδήγηση που μου παρείχε σε όλα τα στάδια της εκπόνησης της παρούσας διπλωματικής εργασίας όπως και τους συνεπιβλέποντες καθηγητές κ. Μπέλλα Νικόλαο και κ.Βασιλακόπουλο Μιχαήλ. Επιπλέον, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στους γονείς μου και την αδερφή μου, που στάθηκαν δίπλα μου καθ'όλη την διάρκεια των σπουδών και χωρίς αυτούς τίποτα από όσα έχω καταφέρει μέχρι σήμερα δε θα ήταν πραγματικότητα. Τέλος, ένα μεγάλο ευχαριστώ στους ανθρώπους που συνάντησα σε αυτή την πορεία και στον πολύτιμο μου άνθρωπο, Θανάση.

Πρόλογος

Η παρούσα εργασία εκπονήθηκε ως το τελευταίο βήμα για την απόκτηση διπλώματος και την ολοκλήρωση των σπουδών μου στο τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας στην πόλη του Βόλου υπό την επίβλεψη του Αναπληρωτή Καθηγητή κ. Γεράσιμου Ποταμιάνου.

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος πινάκων

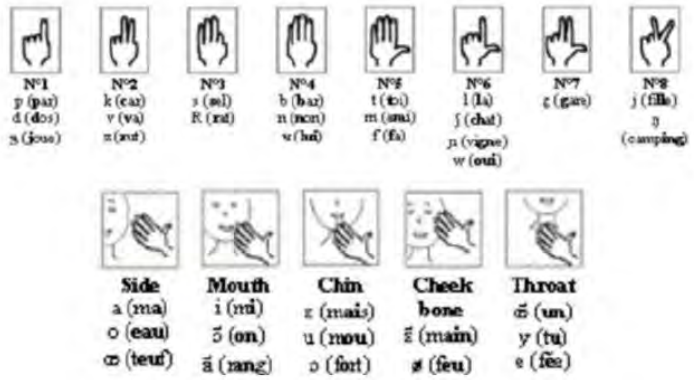
Κεφάλαιο 1

Εισαγωγή

1.1 Τι είναι η Σιωπηρή Ομιλία (Cued Speech)

Η σιωπηρή ομιλία επινοήθηκε το 1966 στην Ουάσιγκτον από τον Dr. R. Orin Cornett, αργότερα μεταφέρθηκε στο Ηνωμένο Βασίλειο από τον Winifred Tumim και προωθήθηκε σε πολλές χώρες με την αέναη βοήθεια του June Dixon-Millar, με αποτέλεσμα να απαριθμεί, πλέον, 60 διαλέκτους [26].

Αποτελείται από ένα χειροκίνητο σύστημα οκτώ διαφορετικών χειρονομιών, οι οποίες κωδικοποιούν τα σύμφωνα και η τοποθέτησή τους σε πέντε θέσεις γύρω από το πρόσωπο (λαιμός, ζυγωματικά, πιγούνι, στόμα και μια θέση εκτός προσώπου) κωδικοποιούν τα φωνήεντα. Ο συνδυασμός τους, λοιπόν, με τις κινήσεις των χειλιών αποσαφηνίζει τα 34 συνολικά φωνήματα της γαλλικής γλώσσας και βοηθά στην κατανόηση του προφορικού λόγου από τα άτομα με προβλήματα ακοής [26].



Σχήμα 1.1: Σύστημα Χειρονομιών Σιωπηρής Ομιλίας (σχήμα από [3])

1.2 Χρησιμότητα Σιωπηρής Ομιλίας

Έχει παρατηρηθεί ότι ένα ποσοστό της τάξης του 70% των κωφών παιδιών εγκαταλείπουν το δημοτικό σχολείο με εξαιρετικά χαμηλές επιδόσεις στις δεξιότητες ανάγνωσης. Το γεγονός αυτό καθιστά αναγκαία την εκμάθηση της σιωπηρής ομιλίας από το οικογενειακό περιβάλλον, τους δασκάλους, τους ειδικούς παράλληλης στήριξης (πχ. ψυχολόγοι) και το ίδιο το παιδί. Τα αποτελέσματα της χρήσης της εντυπωσιάζουν, καθώς τα κωφά παιδιά μπορούν να επιτύχουν ανάλογες δυνατότητες ανάγνωσης με παιδιά που ακούν [26].

Τα οφέλη της είναι πολυδιάστατα, καθώς συμβάλλει στην αμφίδρομη επικοινωνία κωφών ανθρώπων και μη, βοηθά στην εκμάθηση ξένων γλωσσών ενώ η εκπαίδευση των ατόμων στο σύστημα της Σιωπηρής Ομιλίας απαιτεί μόνο 10 με 20 ώρες [26].

1.3 Σιωπηρή Ομιλία και Υπολογιστές

Η εξέλιξη του επιστημονικού πεδίου της όρασης υπολογιστών έχει επιφέρει πολλές δυνατότητες στην επίλυση κοινωνικών προβλημάτων, μέσα από την κατανόηση ψηφιακών εικόνων και βίντεο, εξάγοντας πληροφορίες και δεδομένα για επεξεργασία. Προσομοιώνοντας έτσι την ανθρώπινη όραση, αντίστοιχα, μπορούμε να χρησιμοποιήσουμε αυτή τη διαδικασία για την αναγνώριση της σιωπηρής ομιλίας.

- Με τη χρήση μηχανικής μάθησης, για την επεξεργασία του οπτικού υλικού – βίντεο, θα χρειαστούμε τη βοήθεια των συνελικτικών νευρωνικών δικτύων - Convolutional Neural Networks (CNNs) – τα οποία είναι κατάλληλα για την επεξεργασία των εικόνων των χεριών και χειλιών και την κατηγοριοποίησή τους.

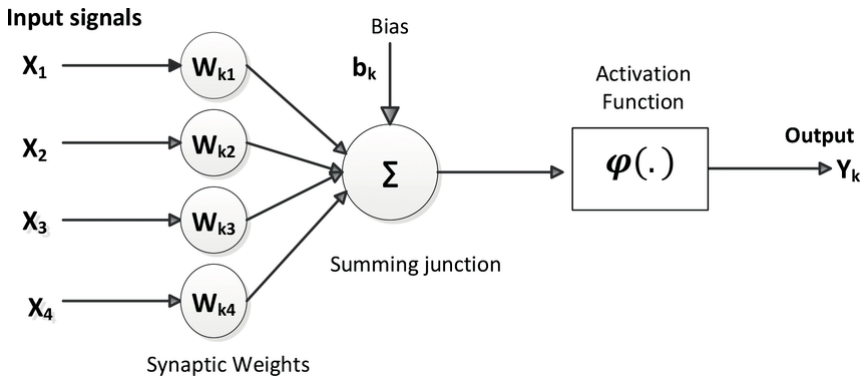
- Για την επεξεργασία του ακουστικού υλικού (audio signal), υπάρχουν αρκετά εργαλεία, όπως το Hidden Markov Model Toolkit (HTK) [1] και το Kaldi [2] που κατασκευάζουν και χειρίζονται μοντέλα μαρκοβιανών αλυσίδων. Με τη χρήση αυτών, εξάγονται πληροφορίες για τα χρονικά διαστήματα στα οποία προφέρονται οι εξαρτημένες μεταβλητές (targets), που θα χρησιμοποιηθούν για την εκπαίδευση των νευρωνικών δικτύων για τα χείλη.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο Νευρωνικών Δικτύων

2.1 Λειτουργία Νευρωνικών Δικτύων

Τα νευρωνικά είναι βιολογικά εμπνευσμένα δίκτυα, που επιτρέπουν σε ένα υπολογιστή να μάθει από δεδομένα. Παρέχουν τις καλύτερες λύσεις για προβλήματα όπως η αναγνώριση εικόνας και ομιλίας. Αποτελούνται από στρώματα απλών υπολογιστικών κόμβων (νευρώνες), διασυνδεδεμένων μεταξύ τους. Τα μοτίβα εισέρχονται στο δίκτυο από το στρώμα εισόδου, το οποίο δεν επιτελεί κάποια υπολογιστική πράξη, απλά προωθεί την είσοδο στα επόμενα υπολογιστικά στρώματα. Αυτά, πολλαπλασιάζουν κάθε είσοδό τους με το αντίστοιχο συναπτικό βάρος, ενίοτε προσθέτεται και μια σταθερά (bias) και υπολογίζουν το ολικό άθροισμα των γινομένων. Έπειτα από κάθε υπολογιστικό στρώμα το αποτέλεσμα εισέρχεται στην εκάστοτε συνάρτηση ενεργοποίησης. Το τελικό στρώμα είναι το στρώμα εξόδου, το οποίο παράγει την έξοδο του νευρωνικού δικτύου. Αυτή μπορεί να αναπαριστά ένα πραγματικό αριθμό στόχο (target) ή να αντιπροσωπεύει μια κλάση σε προβλήματα ταξινόμησης [15].



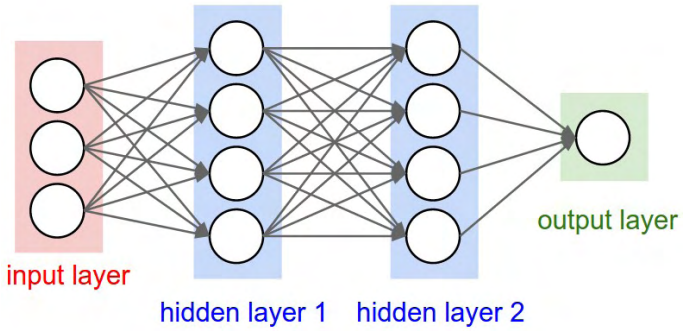
Σχήμα 2.1: Οπτικοποίηση υπολογιστικού νευρώνα (σχήμα από [15])

2.2 Συνελικτικά Νευρωνικά Δίκτυα – Convolutional Neural Networks (CNNs)

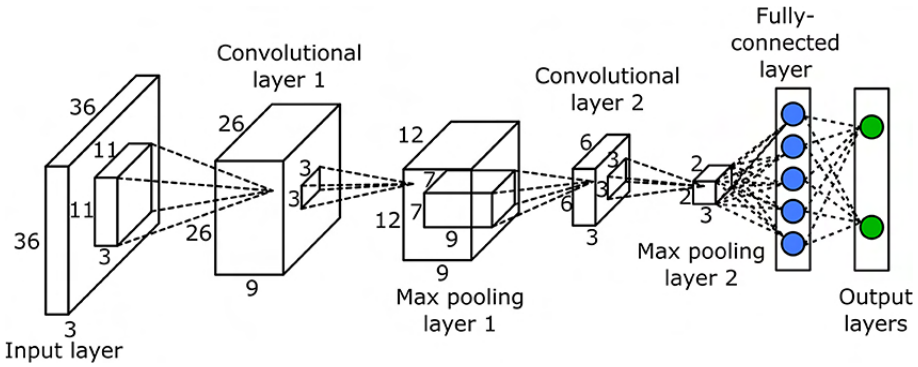
Τα συνελικτικά νευρωνικά δίκτυα (CNNs) έχουν καθιερωθεί ως μια ισχυρή κατηγορία μοντέλων δικτύων για προβλήματα αναγνώρισης εικόνων [31]. Ένα συνελικτικό νευρωνικό δίκτυο είναι σε θέση να καταγράψει με επιτυχία τις χωρικές και προσωρινές εξαρτήσεις σε μια εικόνα μέσω της εφαρμογής φίλτρων πάνω σε αυτή. Έχουν σχεδιαστεί για να λειτουργούν με δομές εισόδου, οι οποίες έχουν ισχυρές χωρικές εξαρτήσεις. Χαρακτηριστικό παράδειγμα των δομών αυτών είναι οι δύο διαστάσεων εικόνες, τύπος δεδομένων ο οποίος έχει παρόμοιες χρωματικές τιμές σε γειτονικά εικονοστοιχεία [12].

Κάθε στρώμα στο συνελικτικό δίκτυο είναι μια τρισδιάστατη δομή πλέγματος, η οποία έχει ύψος, πλάτος και βάθος. Το βάθος ενός στρώματος σε ένα συνελικτικό νευρικό δίκτυο δεν πρέπει να συγχέεται με το βάθος του ίδιου του δικτύου. Όταν χρησιμοποιείται σε ένα μόνο στρώμα αναφέρεται στον αριθμό των καναλιών σε κάθε στρώμα, όπως για παράδειγμα ο αριθμός των κύριων καναλιών (π.χ. μπλε, πράσινο και κόκκινο) στην εικόνα εισόδου ή ο αριθμός των χαρτών χαρακτηριστικών (feature maps) στα κρυφά επίπεδα [15].

Το συνελικτικό νευρωνικό δίκτυο λειτουργεί σαν ένα παραδοσιακό νευρωνικό δίκτυο τροφοδοσίας, εκτός από το ότι οι πράξεις στα στρώματά του είναι χωρικά οργανωμένες με προσεκτικά σχεδιασμένες συνδέσεις μεταξύ των στρωμάτων [15].



Σχήμα 2.2: Νευρωνικό Δίκτυο (σχήμα από [23])



Σχήμα 2.3: Συνελικτικό Νευρωνικό Δίκτυο (σχήμα από [16])

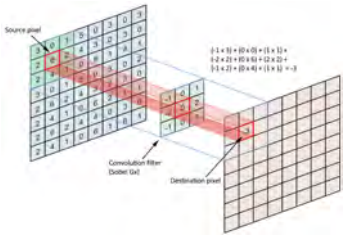
Υπάρχουν τρεις τύποι στρωμάτων σε ένα συνελικτικό νευρωνικό δίκτυο:

1. Convolution
2. Pooling
3. Πλήρως συνδεδεμένα- Fully Connected

2.2.1 Συνελικτικό Στρώμα - Convolution Layer

Λειτουργία συνέλιξης

Στα συνελικτικά δίκτυα, κύρια δομή είναι τα φίλτρα, τα οποία οργανώνονται σε τρισδιάστατες δομές αποτελούμενα από αριθμητικούς πίνακες. Τα φίλτρα είναι συνήθως τετράγωνα και αρκετά μικρότερα από τα στρώματα στα οποία εφαρμόζονται και διάστασης $F_q \times F_q \times d_q$, όπου για $q > 1$ συνήθως αναφέρεται στο χάρτη χαρακτηριστικών, που είναι ανάλογος των τιμών στα κρυφά στρώματα. Το βάθος ενός φίλτρου είναι πάντα ίσο με τον αριθμό των φίλτρων που εφαρμόζονται στο στρώμα.



Σχήμα 2.4: Λειτουργία συνέλιξης (σχήμα από [9])

Το φίλτρο εφαρμόζεται σε όλη την εικόνα διάστασης $L_q \times B_q \times d_q$ και υπολογίζει το εσωτερικό γινόμενο των εικονοστοιχείων της περιοχής που καλύπτει. Ένα συνελικτικό επίπεδο (convolutional layer) είναι ουσιαστικά ένα σύνολο από νευρώνες που εκτελούν συνέλιξη των φίλτρων που έχουν προκαθοριστεί με την εικόνα-διάνυσμα που δέχονται στην είσοδο [15].

Ας ορίσουμε τώρα τη μαθηματική προσέγγιση της λειτουργίας των συνελικτικών δικτύων για το p-οστό φίλτρο στο q-οστό στρώματα που έχει παραμέτρους που δηλώνονται με ένα τρισδιάστατο τένσορα $W^{(p,q)} = [w(p,q)_{ijk}]$.
Οι δείκτες i, j, k δείχνουν τις θέσεις κατά μήκος του ύψους, του πλάτους και του βάθους του φίλτρου. Οι χάρτες χαρακτηριστικών στο q-οστό στρώμα αντιπροσωπεύονται από τον τρισδιάστατο τένσορα $H^{(q)} = [h^{(q)}_{ijk}]$.
Όταν η τιμή του q είναι 1, η ειδική περίπτωση που αντιστοιχεί στο $H(1)$ αντιπροσωπεύει απλώς το στρώμα εισόδου. Στη συνέχεια, οι συνελικτικές πράξεις από το q-οστό στρώμα στο q+1-οστό στρώμα ορίζονται ως εξής [15]:

$$h^{(q+1)}_{i,j,k} = \sum_{r=1}^{F_q} \sum_{s=1}^{F_q} \sum_{k=1}^{d_q} w^{(p,q)}_{r,s,k} h^{(q)}_{i+r-1,j+s-1,k} \tag{2.1}$$

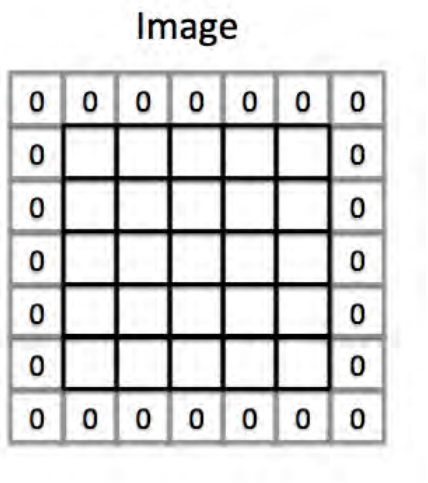
$$\forall i \in \{1, \ldots, L_q - F_{q+1}\}, \forall j \in \{1, \ldots, B_q - F_{q+1}\}, \forall i \in \{1, \ldots, d_{q+1}\}$$

Padding

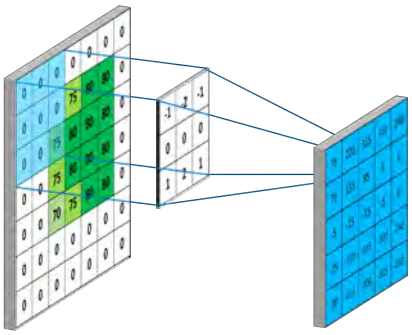
Η λειτουργία συνέλιξης μειώνει το μέγεθος του $(q+1)$- οστού στρώματος σε σχέση με το μέγεθος του q-οστού κατά $L_{q+1} = L_q - F_q + 1$ ως προς το πλάτος και $B_{q+1} = B_q - F_q + 1$ ως προς το ύψος. Αυτός ο τύπος μείωσης του μεγέθους δεν είναι επιθυμητός, διότι τείνει να χάσει κάποιες πληροφορίες κατά μήκος των ορίων της εικόνας (ή του χάρτη χαρακτηριστικών, στην περίπτωση των κρυφών επιπέδων).

Αυτό το πρόβλημα μπορεί να επιλυθεί χρησιμοποιώντας το padding. Στο padding, προσθέτουμε $(F_q - 1)/2$ εικονοστοιχεία (pixels) αρχικοποιημένα σε μηδέν, γύρω από τα όρια του χάρτη χαρακτηριστικών για να διατηρήσουμε το χωρικό αποτύπωμα.

Κατά μία έννοια, αυτό που επιτρέπει το padding είναι η συνέλιξη με ένα τμήμα του φίλτρου εκτός των ορίων του στρώματος και στη συνέχεια ο υπολογισμός του εσωτερικού γινομένου μόνο επί του τμήματος του στρώματος όπου ορίζονται οι τιμές. Αυτός ο τύπος επένδυσης αναφέρεται ως half-padding επειδή (σχεδόν) το ήμισυ του φίλτρου βγαίνει από όλες τις πλευρές της εικόνας εισόδου στην περίπτωση όπου το φίλτρο τοποθετείται στην ακραία του χωρική θέση κατά μήκος των άκρων [15].



Σχήμα 2.5: Padding (σχήμα από [7])



Σχήμα 2.6: Half Padding (σχήμα από [21])

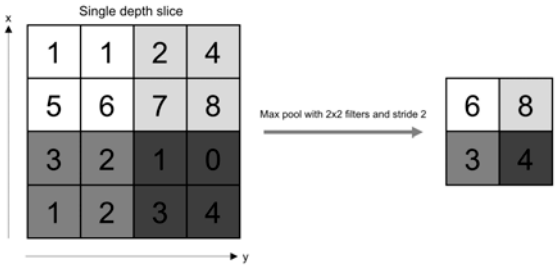
Ολίσθηση Φίλτρου-Strides

Δεν είναι απαραίτητο το φίλτρο να διαπεράσει όλες τις θέσεις του στρώματος. Τα strides, λοιπόν, αντιπροσωπεύουν το βήμα ολίσθησης του φίλτρου πάνω στην εικόνα. Συνήθως, το βήμα ολίσθησης είναι μεγαλύτερο του ένα, διότι συμβάλλει στην μείωση της πολυπλοκότητας του χάρτη χαρακτηριστικών.

2.2.2 Συγκεντρωτικό Στρώμα - Pooling Layer

Η λειτουργία *pooling* λειτουργεί σε μικρές περιοχές πλέγματος μεγέθους $P_q \times P_q$ σε κάθε στρώμα, και παράγει ένα άλλο στρώμα με το ίδιο βάθος (αντίθετα από τα φίλτρα). Για κάθε τετραγωνική περιοχή μεγέθους $P_q \times P_q$ σε κάθε d_q, επιστρέφεται το μέγιστο αυτών των τιμών. Αυτή η προσέγγιση αναφέρεται ως *maxpooling*. Αντί για την μέγιστη τιμή θα μπορούσε να λαμβάνει το μέσο όρο των τιμών αυτής της περιοχής, η προσέγγιση αυτή ονομάζεται average pooling [15].

Εάν χρησιμοποιείται ένα βήμα 1, τότε αυτό θα δημιουργήσει ένα νέο στρώμα μεγέθους $(L_q - P_q + 1) \times (B_q - P_q + 1) \times d_q$. Ωστόσο, είναι πιο συνηθισμένο να χρησιμοποιείται ένα βήμα $S_q > 1$ στο *pooling*. Σε αυτές τις περιπτώσεις, το μήκος του νέου στρώματος θα είναι $(L_q - P_q)/S_q + 1$ και το πλάτος θα είναι $(B_q - P_q)/S_q + 1$. Συνεπώς, το *pooling* μειώνει δραστικά τις χωρικές διαστάσεις του κάθε χάρτη ενεργοποίησης (activation map) [15].

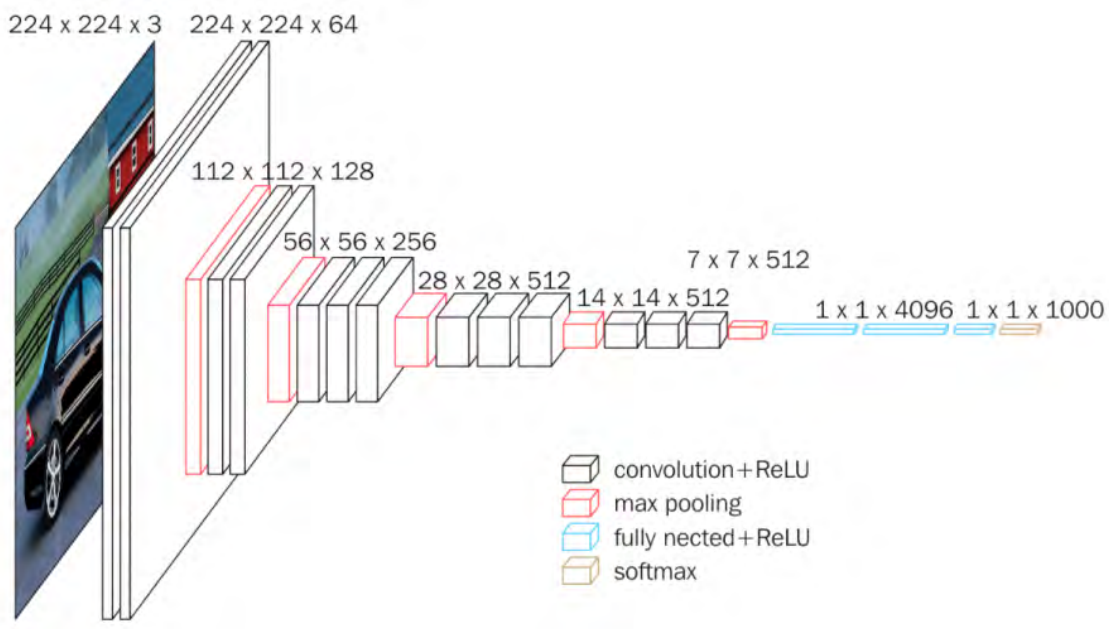


Σχήμα 2.7: Max pooling (σχήμα από [25])

2.2.3 Πλήρως Συνδεδεμένα Στρώματα - Fully Connected Layers

Στα πλήρως συνδεδεμένα στρώματα, κάθε νευρώνας συνδέεται με όλους τους κόμβους του προηγούμενου στρώματος και λειτουργεί ως ένα απλό *feedforward* δίκτυο. Συνήθως, τα στρώματα αυτά χρησιμοποιούνται για την αύξηση της ισχύος των υπολογισμών προς το τέλος της αρχιτεκτονικής και έχουν αρκετά μεγάλο αριθμό παραμέτρων.

Παρακάτω ακολουθεί η VGG16 αρχιτεκτονική δικτύου που περιλαμβάνει το συνδυασμό των προαναφερθέντων στρωμάτων [15].



Σχήμα 2.8: VGG16 αρχιτεκτονική (σχήμα από [28])

2.3 Συναρτήσεις Ενεργοποίησης

Οι συναρτήσεις ενεργοποίησης ή μεταφοράς λαμβάνουν την έξοδο κάθε κόμβου του νευρωνικού δικτύου και εφαρμόζουν μια μαθηματική λειτουργία, η οποία επιλέγεται από τον σχεδιαστή του δικτύου. Συνήθεις επιλογές είναι η σιγμοειδής συνάρτηση *sigmoid*, *tahn*, *softmax*, *ReLU*, *LeakyReLU*, *Softplus* και *Exponential Linear Unit (ELU)* [14].

2.4 Βελτιστοποιητές - Optimizers

Οι αλγόριθμοι βελτιστοποίησης μας βοηθούν να ελαχιστοποιήσουμε την συνάρτηση σφάλματος $E(x)$. Τα βάρη (w) και τα διανύσματα bias (b) του νευρωνικού δικτύου είναι οι εσωτερικές παράμετροι εκπαίδευσης, που χρησιμοποιούνται για τον υπολογισμό των τιμών εξόδου και ενημερώνονται προς την κατεύθυνση της βέλτιστης λύσης. Δηλαδή ελαχιστοποιούν το σφάλμα κατά την εκπαίδευση του δικτύου. Η επιλογή του αλγορίθμου βελτιστοποίησης επηρεάζει σε μεγάλο βαθμό τη διαδικασία εκπαίδευσης και τις εξόδους του μοντέλου μας [30]. Οι αλγόριθμοι βελτιστοποίησης χωρίζονται σε δύο κλάσεις:

1. Αλγόριθμοι βελτιστοποίησης πρώτης τάξης
   Αυτοί οι αλγόριθμοι ελαχιστοποιούν μια συνάρτηση σφάλματος $E(x)$ χρησιμοποιώντας τις τιμές κλίσης σε σχέση με τις παραμέτρους. Ο πιο ευρέως γνωστός αλγόριθμος βελτιστοποίησης πρώτης τάξης είναι ο αλγόριθμος απότομης καθόδου [30].

2. Αλγόριθμοι βελτιστοποίησης δεύτερης τάξης
   Αυτοί οι αλγόριθμοι ελαχιστοποιούν μια συνάρτηση σφάλματος $E(x)$ χρησιμοποιώντας τη δεύτερη παράγωγο ή αλλιώς τον εσσιανό πίνακα (*Hessian*). Η παράγωγος δεύτερης τάξης υποδηλώνει αν η πρώτη παράγωγος αυξάνεται ή μειώνεται, συνεπώς προσδιορίζει την καμπυλότητα της επιφάνειας [30].

2.5 Συναρτήσεις Σφάλματος - Loss Functions

Η συνάρτηση κόστους, απώλειας ή σφάλματος είναι ένα σημαντικό κεφάλαιο στα νευρωνικά δίκτυα. Χρησιμοποιείται για την μέτρηση της ασυνέπειας μεταξύ της προβλεπόμενης και πραγματικής τιμής, για την αξιολόγηση των βαρών του νευρωνικού δικτύου. Αυτό που προσπαθούμε να επιδιώξουμε είναι η ελαχιστοποίηση της συνάρτησης σφάλματος για την βέλτιστη εκπαίδευση του δικτύου [4]. Υπάρχουν τρία είδη συναρτήσεων σφάλματος:

1. Παλινδρομικές
2. Δυαδικής ταξινόμησης
3. Ταξινόμησης πολλαπλών κατηγοριών

Κάτωθι, περιγράφονται κάποιες γνωστές συναρτήσεις:

Μέσο τετραγωνικό σφάλμα - Mean Square Error (MSE)

Η συνάρτηση μέσου τετραγωνικού σφάλματος ή τετραγωνική χρησιμοποιείται για γραμμικά παλινδρομικά προβλήματα, δηλαδή σε προβλήματα όπου η πρόβλεψη είναι ένας πραγματικός αριθμός. Ο τύπος της ορίζεται ως εξής [29]:

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2 \tag{2.2}$$

Όπου y η πραγματική τιμή, $\hat{y}$ η προβλεπόμενη τιμή και n το σύνολο των δεδομένων.

Cross Entropy

Η συνάρτηση αυτή, χρησιμοποιείται σε προβλήματα, όπου η πρόβλεψη είναι είτε η κλάση 1, είτε η κλάση 0. Η cross entropy μετρά την απόκλιση μεταξύ της πραγματικής και της προβλεπόμενης κατανομής πιθανοτήτων. Αν είναι μεγάλη, σημαίνει ότι η διαφορά μεταξύ των δύο κατανομών είναι μεγάλη, ενώ αν είναι μικρή, αυτό σημαίνει ότι οι δύο κατανομές είναι παρόμοιες μεταξύ τους. Για πολλές κλάσεις, την ονομάζουμε multi-class cross entropy και ορίζεται με τον τύπο [20]:

$$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n}[y^{(i)}\log\hat{y}^{(i)} + (1-y^{(i)})\log(1-\hat{y}^{(i)})] \tag{2.3}$$

2.6 Υπερπροσαρμογή - Overfitting

Το πιο κοινό πρόβλημα που συναντά κανείς στην εκπαίδευση νευρωνικών δικτύων είναι το φαινόμενο της υπερπροσαρμογής. Τα δεδομένα του συνόλου εκπαίδευσης που μελετούνται πολλές φορές έχουν ένα βαθμό σφάλματος ή θόρυβο. Η υπερπροσαρμογή είναι ένα σφάλμα μοντελοποίησης, που έχει ως αποτέλεσμα η συνάρτηση που περιγράφει αυτό το μοντέλο να παίρνει περίπλοκες μορφές με στόχο την επεξήγηση αυτών των δεδομένων. Έτσι, προσπαθώντας να καταστήσει το μοντέλο κατάλληλο σε ελαφρώς ανακριβή δεδομένα μπορεί να μολύνει το μοντέλο με σημαντικά λάθη και να μειώσει την προβλεπτική ισχύ του.

Για να μετρήσουμε την ύπαρξη υπερπροσαρμογής, μπορούμε να χρησιμοποιήσουμε ένα σύνολο δεδομένων (validation set), το οποίο δεν έχει περάσει από τη διαδικασία εκπαίδευσης και να παρατηρήσουμε την απόδοση του. Για να αυξήσουμε την ευελιξία του μοντέλου μας, χρησιμοποιούνται οι παρακάτω τεχνικές:

1. **Κανονικοποίηση - Μηχανισμός Dropout**
   Αυτή η ιδέα είναι πραγματικά πολύ απλή. Για κάθε κόμβο του νευρωνικού μας δικτύου (εκτός εκείνων που ανήκουν στο στρώμα εξόδου) δίνεται η πιθανότητα p να αγνοηθεί προσωρινά στους υπολογισμούς. Η παράμετρος *Hyper* ονομάζεται συχνότητα εγκατάλειψης και πολύ συχνά η προεπιλεγμένη τιμή της είναι ρυθμισμένη στο 0.5. Στη συνέχεια, σε κάθε επανάληψη, επιλέγουμε τυχαία τους νευρώνες, σύμφωνα με την εκχωρηθείσα πιθανότητα. Ως αποτέλεσμα, κάθε φορά εργαζόμαστε με ένα μικρότερο νευρωνικό δίκτυο. Δεδομένου ότι σε κάθε επανάληψη, οποιαδήποτε τιμή εισόδου μπορεί να εξαλειφθεί τυχαία, ο νευρώνας

προσπαθεί να ισορροπήσει τον κίνδυνο και να μην ευνοήσει κάποια από τα χαρακτηριστικά. Ως αποτέλεσμα, οι τιμές στον πίνακα βαρών να κατανέμονται πιο ομοιόμορφα [8].

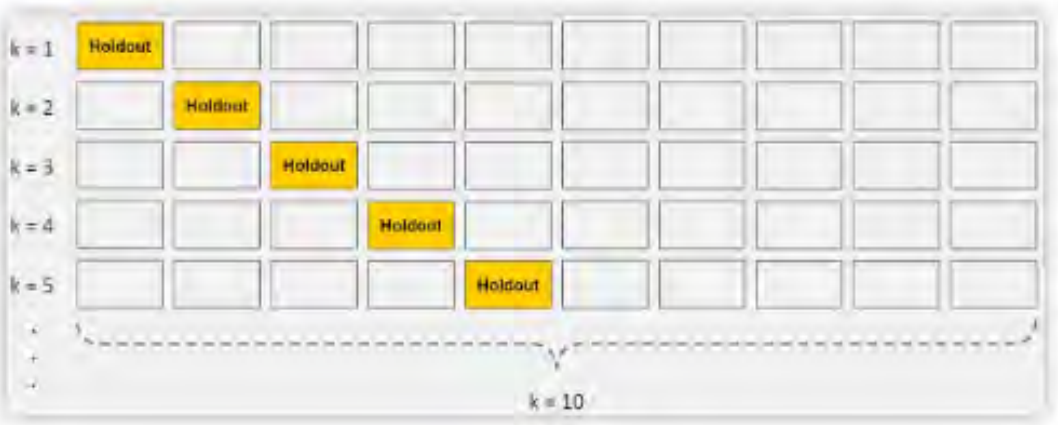
2. **Περισσότερα δεδομένα**
   Δεν είναι η πιο αποτελεσματική μέθοδος, αλλά η εκπαίδευση με περισσότερα δεδομένα μπορεί να βοηθήσει τους αλγόριθμους να ανιχνεύσουν καλύτερα το σήμα. Φυσικά, αυτό δεν συμβαίνει πάντα. Αν προσθέσουμε μόνο πιο θορυβώδη δεδομένα, αυτή η τεχνική δεν θα βοηθήσει. Γι 'αυτό θα πρέπει πάντα να διασφαλίζουμε ότι τα δεδομένα μας είναι καθαρά και συναφή [8].

3. **Μείωση της πολυπλοκότητας δικτύου**
   Ένα μοντέλο μπορεί να υπερπροσαρμοστεί σε ένα σύνολο δεδομένων επειδή έχει την επαρκή ικανότητα για να το κάνει. Η μείωση της ικανότητας του μοντέλου μειώνει την πιθανότητα το μοντέλο να υπερπροσαρμοστεί. Η χωρητικότητα ενός μοντέλου νευρωνικού δικτύου, η πολυπλοκότητά του, ορίζεται τόσο από τη δομή του ως προς τους κόμβους, τα στρώματα όσο και από τις παραμέτρους ως προς τα βάρη του. Ως εκ τούτου, μπορούμε να μειώσουμε την πολυπλοκότητα ενός νευρωνικού δικτύου για να μειώσουμε την υπερπροσαρμογή με έναν από τους δύο τρόπους [8]:

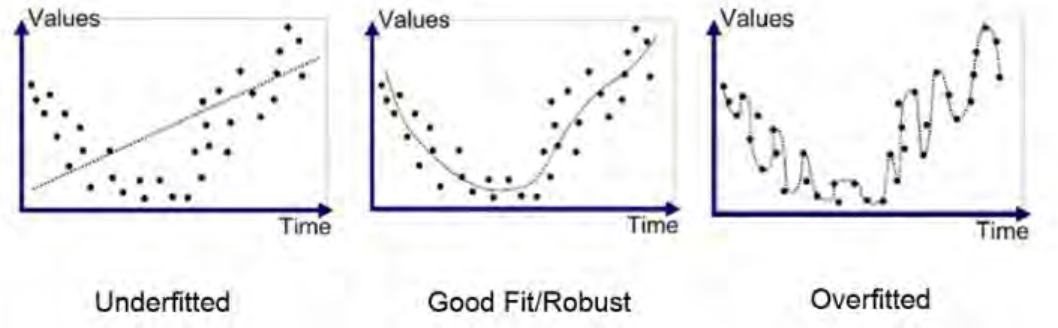
   - Αλλάζοντας τη δομή του δικτύου (αριθμός βαρών).
   - Αλλάζοντας τις παραμέτρους του δικτύου (τιμές των βαρών).

4. **Cross validation**
   Μια έξυπνη ιδέα, ώστε να χωρίσουμε το αρχικό σύνολο δεδομένων προς εκπαίδευση σε μικρά train-test σύνολα για τον συντονισμό του μοντέλου. Αυτό μας επιτρέπει να διατηρήσουμε το σετ δοκιμών (test set) ως ένα πραγματικά αόρατο σύνολο δεδομένων για τον έλεγχο του τελικού μοντέλου. Σε μια τυπική k-fold cross validation διαδικασία, έχουμε k υποσύνολα, τα οποία ονομάζονται folds. Στη συνέχεια, ακολουθούμε συνεχώς τον αλγόριθμο στα k-1 folds, ενώ χρησιμοποιούμε το υπόλοιπο fold ως δοκιμαστικό σετ (που ονομάζεται fold holdout) [8].

Εκτός από το πρόβλημα της υπερπροσαρμογής υπάρχει και η υποπροσαρμογή (underfitting), η οποία είναι ακριβώς το αντίθετο πρόβλημα από αυτό που αναλύθηκε παραπάνω. Κύριο χαρακτηριστικό της είναι η αποτυχία εκμάθησης των σχέσεων μεταξύ των δεδομένων προς εκπαίδευση [22].



Σχήμα 2.9: Cross Validation διαδικασία (σχήμα από [8])



Σχήμα 2.10: Παραδείγματα μοντελοποίησης συναρτήσεων με υποπροσαρμογή, ευρωστία και υπερπροσαρμογή (σχήμα από [5])

Κεφάλαιο 3

Βάση Δεδομένων

3.1 Περιεχόμενα βάσης δεδομένων

Στη βάση δεδομένων έχει βιντεοσκοπηθεί ένας επαγγελματίας διερμηνέας σιωπηρής ομιλίας, ο οποίος προφέρει 238 γαλλικές προτάσεις που επαναλαμβάνονται διπλά, ένα σύνολο δηλαδή 476 προτάσεων. Αποτελείται από έγχρωμες εικόνες από βίντεο από το άνω μέρος του σώματος με ρυθμό δειγματοληψίας κάθε στιγμιότυπου τα 50 fps και ανάλυση 720x576 pixels. Ενώ, ως δεδομένο έχουμε την περιγραφή της γαλλικής γλώσσας μέσω 34 φωνημάτων (20 συμφώνων και 14 φωνηέντων). Συνολικά, η βάση περιείχε [19]:

- video: εικόνες κατηγοριοποιημένες σε φακέλους ανά πρόταση
- audio: ακουστικό υλικό από κάθε πρόταση
- corpusmlf.txt: το αρχείο με τα χρονικά διαστήματα αρχής και τέλους της προφοράς κάθε φωνήματος, όπως έχει εξαχθεί απο το HTK
- phonelist.txt: η λίστα με τα 34 συνολικά φωνήματα
- prompt.txt: το αρχείο με τις προτάσεις γραπτώς

3.2 Προ-επεξεργασία Δεδομένων

Για την αποτελεσματική εκπαίδευση των νευρωνικών που θα χρειαστούν στην υλοποίηση, προαπαιτούμενη είναι η προεπεξεργασιά του συνόλου δεδομένων. Αρχικά, εξάγουμε για κάθε φωτογραφία την περιοχή ενδιαφέροντος - Region of Interest (ROI), στην προκειμένη περίπτωση αυτά είναι τα χείλη και τα χέρια και για την υλοποίηση που χρησιμοποιείται στην συγκεκριμένη διπλωματική. Όλες οι εικόνες εισόδου μετατρέπονται σε κλίμακα του γκρίζου (grayscaled) και ανάλυση 64x64 pixels.

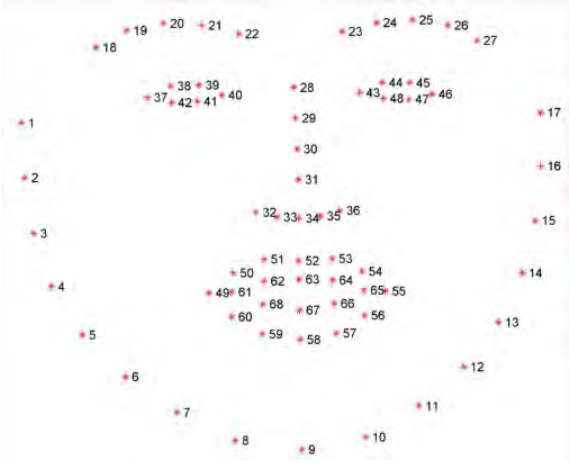
3.2.1 Ανίχνευση περιοχής χειλιών

Για την εξαγωγή των χειλιών είναι απαραίτητη η ανίχνευση των σημείων του προσώπου (facial landmarks), η οποία πραγματοποιείται με την διαδικασία δύο βημάτων:

1. Εντοπισμός προσώπου (με OpenCV's Haar cascades ή προ-εκπαιδευμένο ανιχνευτή αντικείμενων HOG + Linear SVM ή deep learning αλγορίθμους) όπου λαμβάνουμε τις συντεταγμένες (x,y) οριοθέτησης του προσώπου.

2. Εντοπισμός βασικών δομών του προσώπου (στόμα, φρύδια, μύτη, μάτια, σαγόνι) που περιέχεται στην βιβλιοθήκη dlib της Python. Η μέθοδος χρησιμοποιεί ένα εκπαιδευμένο σύνολο με εικόνες επισημανσμένες χειροκίνητα, καθορίζοντας τις συντεταγμένες (x,y) κάθε δομής προσώπου και την πιθανότητα απόστασης των ζευγαριών pixels εισόδου [24].

Το τελικό αποτέλεσμα είναι ότι ο ανιχνευτής μπορεί να χρησιμοποιηθεί για τον εντοπισμό σημείων προσώπου σε πραγματικό χρόνο.



Σχήμα 3.1: Οπτικοποίηση σημείων προσώπου (σχήμα από [24])

Επομένως, μπορούμε να λάβουμε τις συντεταγμένες (x,y) που αφορούν τα χείλη γνωρίζοντας ότι τα σημεία που τα αντιπροσωπεύουν κυμαίνονται στις τιμές [48,68]. Τα αποτελέσματα είναι της μορφής:

Σχήμα 3.2: Σχήμα Χειλιών (α)

Σχήμα 3.3: Σχήμα χειλίων (β)

3.2.2 Εύρεση ετικετών για εκπαίδευση νευρωνικού δικτύου χειλιών

Επεξεργασία αρχείου φωνημάτων

Για την εκπαίδευση των εικόνων των χειλιών αναγκαία είναι η εξαγωγή των εξαρτημένων μεταβλητών από το αρχείο κειμένου *corpusmlf.txt*. Σε αυτό περιέχονται, όπως προανέφερα οι χρόνοι προφοράς κάθε φωνήματος σε 10^{-7} του δευτερολέπτου. Το αρχείο είναι της μορφής:

```
#!MLF!#
"*/001-1.lab"
0 2600000 __
2600000 3150000 m
3150000 4100000 a
4100000 5600000 s^
5600000 6454818 q
6454818 7211886 m
7211886 9050000 i
9050000 9750000 z
9750000 10710460 e^
10710460 11650000 r
11650000 12550000 u
12550000 14250000 s
14250000 15050000 i
15050000 18450000 __
.
```

Σχήμα 3.4: Διάταξη αρχείου στόχων

Προσέγγιση 1η

Δεδομένου του ρυθμού δειγματοληψίας των στιγμιοτύπων στα 50 *fps* γνωρίζουμε ότι η περίοδος με την οποία οι φωτογραφίες λαμβάνονται στο βίντεο είναι στα 0.02 του δευτερολέπτου. Με αυτόν τον τρόπο, για κάθε εγγραφή φωνήματος στο αρχείου κείμενου, αφαιρούσα τον χρόνο λήξης με τον χρόνο έναρξης της προφοράς, στρογγυλοποιώντας προς τα πάνω και διαιρούσα με την περίοδο. Η διαδικασία αυτή, γινόταν για την αντιστοίχιση κάθε φωνήματος με το πλήθος των φωτογραφιών του βίντεο, που αναφέρονταν σε αυτό το φώνημα.

Ωστόσο, παρατήρησα ότι για κάθε φάκελο το συνολικό πλήθος των φωτογραφιών που αντιστοιχίζονταν δεν ήταν ισόποσο με τον πραγματικό πλήθος των εικόνων μέσα στον φάκελο. Αυτό, οφείλονταν στο γεγονός ότι το 50 *fps* ήταν κάπως "ιδανικό" και κάθε φάκελος είχε το δικό του *fps*, επομένως έπρεπε να βρεθεί μια εναλλακτική μέθοδος για την αντιστοίχιση.

Προσέγγιση 2η

Σε αυτή την περίπτωση, λοιπόν, ξεκίνησα με την εύρεση του τρέχοντος *fps* για κάθε φάκελο, διαιρώντας τον αριθμό των πραγματικών εικόνων στο φάκελο με το χρόνο λήξης του τελευταίου φωνήματος στο φάκελο, δηλαδή το συνολικό χρόνο προφοράς της πρότασης. Επομένως, είχα την τρέχουσα περίοδο λήψης (T) κάθε εικόνας. Στη συνέχεια, για την αντιστοίχιση των στόχων (targets) σύγκρινα τον εκάστοτε χρόνο λήξης κάθε φωνήματος με την τρέχουσα περίοδο και κατηγοριοποιούσα ανάλογα το πλήθος των φωτογραφιών με τον πραγματικό τους στόχο.

Πάλι όμως, παρατήρησα ότι ενώ η μεθοδολογία αυτή λειτουργούσε άψογα μέχρι ένα συγκεκριμένο φάκελο, μετά χαλούσε και εξαρτημένες μεταβλητές δεν αντιστοιχίζονταν σωστά. Το φαινόμενο

αυτό, υπήρχε λόγω του ότι μέσα στο αρχείο κειμένου υπήρχαν ελάχιστα φωνήματα που προφέρονταν σε χρόνο μικρότερο της περιόδου και δεν προλάβαιναν να αντιστοιχιστούν με κάποια από τις εικόνες των φακέλων. Το πρόβλημα αυτό επιλύθηκε με την αφαίρεσή τους από τη συνολική βάση δεδομένων.

Κατηγοριοποίηση φωνημάτων σε κλάσεις

Κάποια από τα σύμφωνα προφέρονται με τον ίδιο τρόπο στα χείλη και χωρίζονται σε άφωνα με υποκατηγορίες τα οδοντικά, χειλικά, ουρανικά και τα ημίφωνα με υποκατηγορίες τα υγρά και έν-ρινα. Στη γαλλική γλώσσα επίσης, υπάρχουν ποικίλοι τρόποι προφοράς των φωνηέντων ανάλογα με τον τονισμό τους. Επομένως, χρειάστηκε μια αναδιαμόρφωση των φωνημάτων σε 15 κατηγο-ρίες, συμπεριλαμβανομένης της σιωπής. Παρακάτω ακολουθεί ο πίνακας που κατηγοριοποιεί τα φωνήματα σε κλάσεις ανάλογα με το σχήμα που δημιουργείται στα χείλη κατά την προφορά τους (visemes) [13].

Viseme Class	Σύμφωνα	Φωνήεντα
0	silence	silence
1	p,b,m	-
2	t,d,l,n	-
3	f,v	-
4	k,z∧(g)	-
5	w	-
6	j	-
7	s∧(ch)	-
8	s,z	-
9	-	i
10	-	e∧, e, $e\sim$, q
11	-	x, x∧, $x\sim$
12	-	o, o∼, o∧
13	-	a, a∼
14	-	u, y

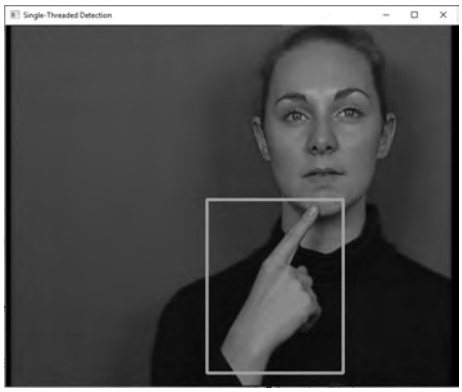
Πίνακας 3.1: Πίνακας κατηγοριοποίησης φωνημάτων

3.2.3 Ανίχνευση περιοχής χεριών

Για την εξαγωγή των χεριών, αρχικά υλοποίησα μεθόδους που βασίζονταν στον εντοπι-σμό του χρώματος του δέρματος και την αφαίρεση του παρασκηνίου. Τα επίπεδα φωτισμού όμως διαφέρουν ανάλογα με τις συνθήκες βιντεοσκόπησης και σε συνδυασμό με το γεγονός ότι κάθε

άνθρωπος έχει διαφορετικό χρώμα δέρματος , τα αποτελέσματα κρίνονταν αναξιόπιστα. Συνεπώς, χρησιμοποίησα ένα ήδη εκπαιδευμένο μοντέλο που βασίζεται στο *TensorflowObjectDetectionAPI*.

Η προσέγγιση που ακολούθησα είναι η φόρτωση του προ-εκπαιδευμένου μοντέλου *frozen_inference_graph.pb*, καθώς και του αντιστοίχου χάρτη ετικετών και το διάβασμα των εικόνων από το σύνολο δεδομένων . Με αυτόν τον τρόπο, εντοπίζονται οι συντεταγμένες ενός ορθογωνίου (left,right,bottom,top), που γραμμοσκιάζεται γύρω από την περιοχή των χεριών [10].



Σχήμα 3.5: Ανίχνευση Χεριού

Σχήμα 3.6: Αποκοπή και εύρεση κέντρου

Κατηγοριοποίηση χειρονομιών σε κλάσεις

Παρακάτω, παρουσιάζονται οι εικόνες έτσι όπως κατηγοριοποιούνται στις οκτώ κλάσεις που χρησιμοποιούνται για την εκπαίδευση του συνελικτικού δικτύου για την εύρεση των χειρονομιών.

Σχήμα 3.7: Χειρονομία 1

Σχήμα 3.8: Χειρονομία 2

Σχήμα 3.9: Χειρονομία 3

Σχήμα 3.10: Χειρονομία 4

Σχήμα 3.11: Χειρονομία 5

Σχήμα 3.12: Χειρονομία 6

Σχήμα 3.13: Χειρονομία 7

Σχήμα 3.14: Χειρονομία 8

3.2.4 Εύρεση κέντρου χεριών

Για την εύρεση των κέντρων των χεριών, αφού πλέον έχω γραμμοσκιασμένα τα ορθογώνια που περικλείουν τα χέρια από την παραπάνω διαδικασία ανίχνευσης της περιοχής ενδιαφέροντος, μπορώ να υπολογίσω τις συντεταγμένες του κέντρου χρησιμοποιώντας τις συντεταγμένες των τεσσάρων κορυφών του ορθογωνίου, με τις παρακάτω εξισώσεις:

$$c_1 = \frac{left + right}{2} \tag{3.1}$$

$$c_2 = \frac{bottom + top}{2} \tag{3.2}$$

3.3 Μετατροπή κατηγορηματικών δεδομένων σε αριθμητικά

Συχνά, τα προβλήματα μηχανικής μάθησης για πραγματικά δεδομένα απαιτούν προετοιμασία των δεδομένων με συγκεκριμένους τρόπους πριν από την δημιουργία ενός μοντέλου μηχανικής μάθησης. Για κατηγορικά δεδομένα, όπως στην περίπτωσή μας, προτείνεται η χρήση της one hot κωδικοποίησης. Τα κατηγορικά δεδομένα είναι μεταβλητές που περιέχουν τιμές ετικέτας και όχι αριθμητικές τιμές, κάθε τιμή αντιπροσωπεύει και μια διαφορετική κλάση. Ο αριθμός των πιθανών τιμών συχνά περιορίζεται σε ένα σταθερό σύνολο. Οι κατηγορικές μεταβλητές ονομάζονται και ονομαστικές [6].

Μερικοί αλγόριθμοι μπορούν να συνεργαστούν άμεσα με τα κατηγορικά δεδομένα. Για παράδειγμα, ένα δέντρο αποφάσεων μπορεί να μάθει απευθείας από κατηγορικά δεδομένα χωρίς να απαιτείται μετασχηματισμός δεδομένων (αυτό εξαρτάται από την συγκεκριμένη υλοποίηση). Πολλοί αλγόριθμοι μηχανικής μάθησης, όμως, δεν μπορούν να λειτουργήσουν απευθείας με τα δεδομένα της ετικέτας. Απαιτούν όλες οι μεταβλητές εισόδου και μεταβλητές εξόδου να είναι αριθμητικές [6]. Η μετατροπή των κατηγορηματικών δεδομένων γίνεται με δυο τρόπους:

1. **Κωδικοποίηση Ακεραίων**
   Σε αυτή την περίπτωση, κάθε τιμή μοναδικής κατηγορίας εκχωρείται ως μια ακέραια τιμή. Για παράδειγμα, η κλάση “side” είναι το 1, η κλάση “mouth” είναι το 2 κ.ο.κ. Αυτό ονομάζεται κωδικοποίηση ετικέτας ή κωδικοποίηση ακέραιων αριθμών.

2. **Κωδικοποίηση One-Hot**
   Στην κωδικοποίηση one-hot, αντί για μία ακέραιη τιμή έχουμε ένα δυαδικό διάνυσμα κωδικοποίησης. Στο δίανυσμα αυτό, κάθε στήλη αντιπροσωπεύει μια κλάση. Το δεδομένο που ανήκει στην j-οστή κλάση θα έχει την τιμή 1 στην j-οστή στήλη και μηδέν στις υπόλοιπες [6].

Στην δική μας υλοποίηση, χρησιμοποιείται ο δεύτερος τρόπος της one hot κωδικοποίησης. Όπως φαίνεται παρακάτω για τους στόχους του νευρωνικού για την εύρεση της θέσης χεριών και των χειρονομιών χρησιμοποιείται η εξής κωδικοποίηση:

Hand Position Class	Side	Mouth	Chin	Cheek	Throat
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	0	1

Πίνακας 3.2: One hot κωδικοποίηση για τις κλάσεις της θέσης χεριών

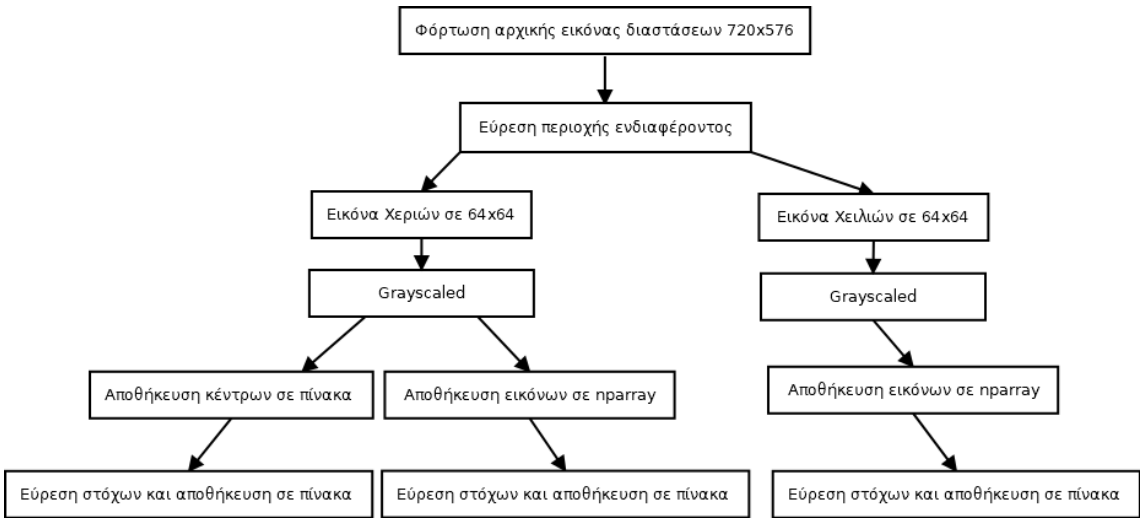
Hand Shape Class	No1	No2	No3	No4	No5	No6	No7	No8
1	1	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0
4	0	0	0	1	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	1	0	0
7	0	0	0	0	0	0	1	0
8	0	0	0	0	0	0	0	1

Πίνακας 3.3: One hot κωδικοποίηση για τις κλάσεις των χειρονομιών

Αντίστοιχα, παρόμοιας μορφής είναι και οι κλάσεις για τα 15 visemes.

3.4 Σύνοψη προ-επεξεργασίας

Η Εικόνα 3.15 εξηγεί σχηματικά την διαδικασία που ακολουθήθηκε για την προ-επεξεργασία των δεδομένων, ώστε να γίνουν οι είσοδοι των νευρωνικών δικτύων που θα εκπαιδευτούν για την ανίχνευση των φωνημάτων. Η αρχιτεκτονική αυτών θα αναλυθεί στο επόμενο κεφάλαιο.



Σχήμα 3.15: Προ-επεξεργασία εικόνων για την τροφοδότηση των νευρωνικών

Κεφάλαιο 4

Προτεινόμενη Υλοποίηση και Αρχιτεκτονικές Νευρωνικών Δικτύων

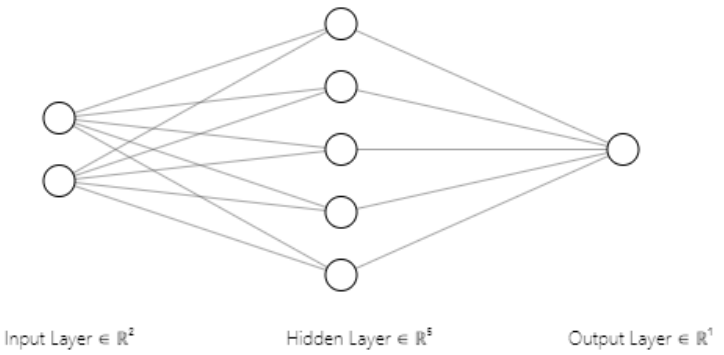
4.1 Εύρεση Θέσης Χεριών

Οι συντεταγμένες του κέντρου, που εξάγονται από την περιοχή ενδιαφέροντος του χεριού χρησιμοποιούνται για την αποκωδικοποίηση των φωνηέντων. Παρακάτω αναλύονται δυο διαφορετικοί τρόποι προσέγγισης για την εύρεση της κλάσης, στην οποία ανήκουν οι συντεταγμένες των κέντρων των χεριών.

4.1.1 Απλό Νευρωνικό Δίκτυο

Μια προσέγγιση είναι ότι οι συντεταγμένες των κέντρων των χεριών εισάγονται σε ένα απλό feed forward νευρωνικό δίκτυο και η έξοδος του δικτύου αντιπροσωπεύει την πιθανότητα ταξινόμησης σε μια από τις πέντε κλάσεις (side, mouth, cheek, chin, throat).

- Το πρώτο στρώμα είναι ένα πλήρως συνδεδεμένο δίκτυο με συνάρτηση μεταφοράς την ReLU.
- Το επόμενο και τελευταίο στρώμα έχει συνάρτηση ενεργοποίησης την softmax [18].



Σχήμα 4.1: Αρχιτεκτονική νευρωνικού δικτύου για εύρεση κέντρου

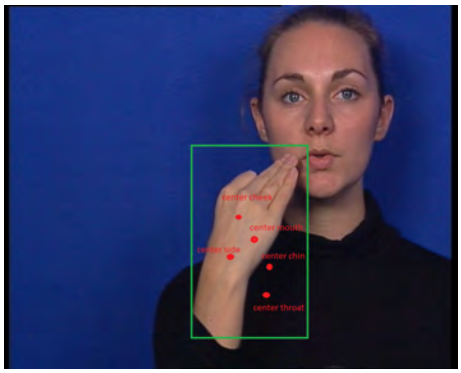
4.1.2 Απόσταση Manhattan

Μια δεύτερη υλοποίηση είναι η ταξινόμηση του κέντρου των χεριών στις πέντε κλάσεις με βάση την απόσταση Manhattan, καθώς το πρόβλημα μας δεν είναι τόσο πολύπλοκο. Manhattan distance ορίζεται ως η απόσταση μεταξύ δύο σημείων που ισούται με το άθροισμα των απόλυτων διαφορών των καρτεσιανών συντεταγμένων [27].

$$d_1(p,q) = \|p-q\|_1 = \sum_{i=1}^{n} |p_i - q_i| \tag{4.1}$$

όπου τα p,q είναι διανύσματα: $p = (p_1, p_2, \ldots, p_n)$ και $q = (q_1, q_2, \ldots, q_n)$.

Επομένως, από το σύνολο δεδομένων, το οποίο είναι διαχωρισμένο σε πέντε φακέλους (Side, Mouth, Cheek, Chin, Throat) με φωτογραφίες της εκάστοτε κλάσης βρίσκω τους μέσους όρους των συντεταγμένων των κέντρων κάθε κλάσης. Δημιουργούνται, δηλαδή 5 συστάδες με κέντρο το μέσο όρο κάθε κλάσης. Όταν, λοιπόν, διαβάζεται μια καινούρια εικόνα, εντοπίζεται το σημείο-κέντρο του χεριού και ταξινομείται στην εγγύτερη κλάση. Παρακάτω, φαίνονται τα κέντρα των κλάσεων που έχουν διαμορφωθεί από το σύνολο των δεδομένων.



Σχήμα 4.2: Απεικόνιση κέντρων κλάσεων για θέση χεριού

4.2 Εύρεση Χειρονομιών

4.2.1 Αρχιτεκτονική Δικτύου

Για την ανίχνευση των οκτώ διαφορετικών χειρονομιών, που θα μας αποκωδικοποιούν την κατηγορία, στην οποία ανήκει το σύμφωνο, δημιουργήθηκε ένα συνελικτικό δίκτυο πολλαπλών στρωμάτων. Το δίκτυο αυτό λαμβάνει ως εισόδους τις εικόνες των χεριών, που έχουν αποκοπεί και αναπροσαρμοστεί σε μέγεθος 64x64x1 και παράγει ως έξοδο ένα διάνυσμα πιθανοτήτων για την ταξινόμηση σε κάθε αντίστοιχη κλάση. Η μεγαλύτερη τιμή είναι και πιθανοτικά η καταλληλότερη. Η αρχιτεκτονική του μοντέλου δικτύου καθορίζεται ως εξής:

- Το πρώτο στρώμα είναι συνελικτικό, με συνάρτηση ενεργοποίησης την ReLU. Σε αυτό εφαρμόζονται οκτώ φίλτρα μεγέθους 7x7 με βήμα 1. Ως αποτέλεσμα το μέγεθος της εικόνας εξόδου μειώνεται σε 21x21.

- Το δεύτερο στρώμα είναι ένα max pooling, που επιστρέφει τις μέγιστες τιμές από ένα παράθυρο μεγέθους 3x3 που ορίζω και σαρώνει την εικόνα, που εν τέλει διαμορφώνεται σε μέγεθος 15x15.

- Τα επόμενα δύο στρώματα είναι ακριβώς ίδια με τα παραπάνω με τελικό μέγεθος εικόνας τα 5x5 pixels.

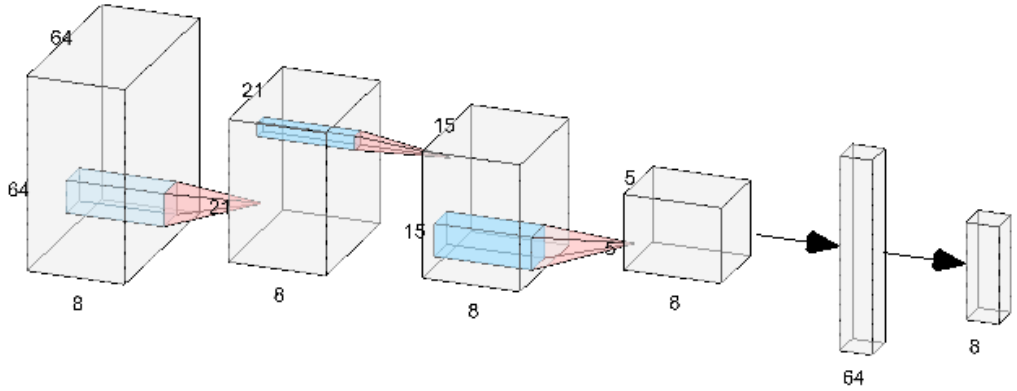
- Στη συνέχεια εφαρμόζω ένα dropout μηχανισμό για την αποφυγή του overfitting με πιθανότητα dropout τα 0.25.

- Το πέμπτο στρώμα είναι είναι ένα πλήρως συνδεδεμένο στρώμα με ReLU συνάρτηση ενεργοποίησης, το οποίο βοηθά στην ισχυροποίηση των υπολογισμών μας, συνδέοντας όλους τους προηγούμενους κόμβους-νευρώνες με τους επόμενους.

- Πάλι ένα μηχανισμό dropout ίδιας πιθανότητας.

- Τέλος, εφαρμόζω επίσης ένα πλήρως συνδεδεμένο στρώμα, μόνο που αυτή τη φορά η συνάρτηση μεταφοράς επιλέγεται η softmax. Με αυτόν τον τρόπο εξασφαλίζω ότι η έξοδος μου θα μου παρέχει την κλάση με την μέγιστη πιθανότητα που κατηγοριοποιείται η χειρονομία [18].

Η εικόνα που ακολουθεί οπτικοποιεί την αρχιτεκτονική του συνελικτικού νευρωνικού δικτύου που παρουσιάστηκε παραπάνω.



Σχήμα 4.3: Αρχιτεκτονική νευρωνικού δικτύου για εύρεση χειρονομιών

4.2.2 Υπερπαράμετροι

Ο καθορισμός των υπερπαραμέτρων πριν από την έναρξη της εκπαίδευσης είναι ένα κρίσιμο σημείο. Οι υπερπαράμετροι είναι οι μεταβλητές που καθορίζουν τη δομή του νευρωνικού δικτύου και οι μεταβλητές που καθορίζουν τον τρόπο κατάρτισής του.

Μέγεθος σακιδίου - Batch size

Μια παραλλαγή εκπαίδευσης του νευρωνικού δικτύου είναι αυτή με τη χρήση σακιδίου (batch). Αυτή η μέθοδος ενημερώνει τα βάρη του δικτύου όχι για κάθε δεδομένο, αλλά για μια ομάδα δεδομένων (batch). Το μέγεθος σακιδίου (batch size) καθορίζει τον αριθμό των δειγμάτων για τα οποία θα γίνει η ενημέρωση των παραμέτρων του εσωτερικού μοντέλου. Ας θεωρήσουμε ένα σακίδιο ως μια for-loop επανάληψη, όπου πάνω από ένα ή περισσότερα δείγματα κάνουμε προβλέψεις. Στο τέλος του σακιδίου, οι προβλέψεις συγκρίνονται με τις αναμενόμενες μεταβλητές εξόδου και υπολογίζεται ένα σφάλμα. Από αυτό το σφάλμα, ο αλγόριθμος ενημέρωσης των παραμέτρων (πχ. αλγόριθμος απότομης καθόδου, RMSprop) χρησιμοποιείται για τη βελτίωση του μοντέλου. Βασικό πλεονέκτημα της χρήσης του batching είναι η ταχύτητα υπολογισμών, καθώς τα βάρη ενημερώνονται ανά batch-size στοιχεία.

Ένα σύνολο δεδομένων μπορεί να χωριστεί σε ένα ή περισσότερα σακίδια (batches). Όταν χρησιμοποιούνται όλα τα δείγματα εκπαίδευσης για τη δημιουργία ενός σακιδίου, ο αλγόριθμος εκμάθησης ονομάζεται batch. Όταν το σακίδιο (batch) έχει το μέγεθος ενός δείγματος, ο αλγόριθμος μάθησης ονομάζεται στοχαστικός. Όταν το μέγεθος του σακιδίου είναι περισσότερο από ένα δείγμα και μικρότερο από το μέγεθος του συνόλου δεδομένων, ο αλγόριθμος μάθησης ονομάζεται mini-batch [11].

Εποχές - Epochs

Ο αριθμός των εποχών είναι ένας υπερπαραμετρικός αριθμός που ορίζει τον αριθμό των επαναλήψεων που ο αλγόριθμος εκμάθησης θα σαρώσει τα δεδομένα μας για να υπολογίσει τα βέλτιστα βάρη. Μια εποχή σημαίνει ότι κάθε δείγμα στο σύνολο δεδομένων κατάρτισης είχε την ευκαιρία να ενημερώσει τις παραμέτρους του εσωτερικού μοντέλου. Μια εποχή αποτελείται από ένα ή περισσότερα σακίδια. Ο αριθμός των εποχών είναι παραδοσιακά μεγάλος, συχνά εκατοντάδες ή χιλιάδες, επιτρέποντας στον αλγόριθμο εκμάθησης να τρέχει μέχρις ότου το σφάλμα από το μοντέλο έχει ελαχιστοποιηθεί επαρκώς [11].

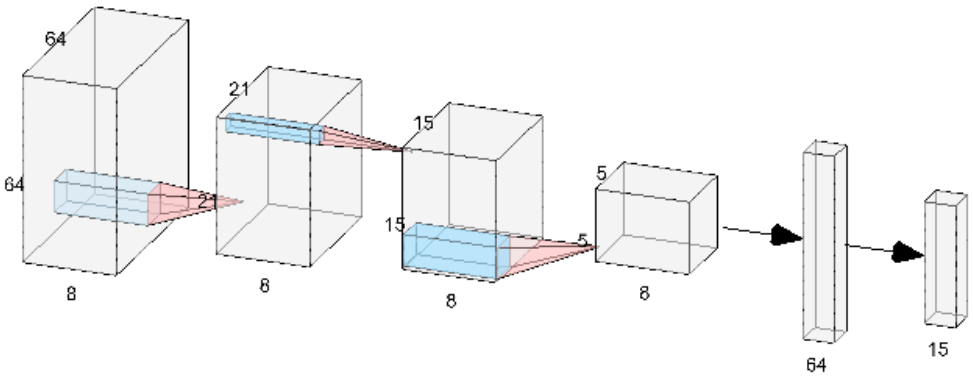
Για την εκπαίδευση αυτού, όρισα ως βελτιστοποιητή τον RMSprop. Επιπλέον, ο ρυθμός εκμάθησης (learning rate) είναι ίσος με 0.0001 και η συνάρτηση σφάλματος ορίζεται ως η categorical cross entropy, αφού τα δεδομένα μου είναι κατηγορικά και αποζητώ διακριτές κλάσεις. Τέλος, εφάρμοσα ένα batch size μεγέθους 2048 εικόνων και αριθμό εποχών 500 [18].

learning rate	0.0001
optimizer	RMSprop
batch size	2048
epochs	600

Πίνακας 4.1: Πίνακας με υπερπαραμέτρους

4.3 Εύρεση Σχήματος Χειλιών

Για την διαδικασία εύρεσης του σχήματος των χειλιών ακολουθήθηκε ακριβώς η παραπάνω αρχιτεκτονική με τη μόνη διαφορά ότι πλέον το δίκτυο πρέπει να κατηγοριοποιεί τις εικόνες σε συνολικά 15 κλάσεις. Η εικόνα που ακολουθεί παρουσιάζει την αρχιτεκτονική που ακολουθήθηκε.



Σχήμα 4.4: Αρχιτεκτονική νευρωνικού δικτύου για εύρεση χειλιών

Κεφάλαιο 5

Αποτελέσματα και Συζήτηση

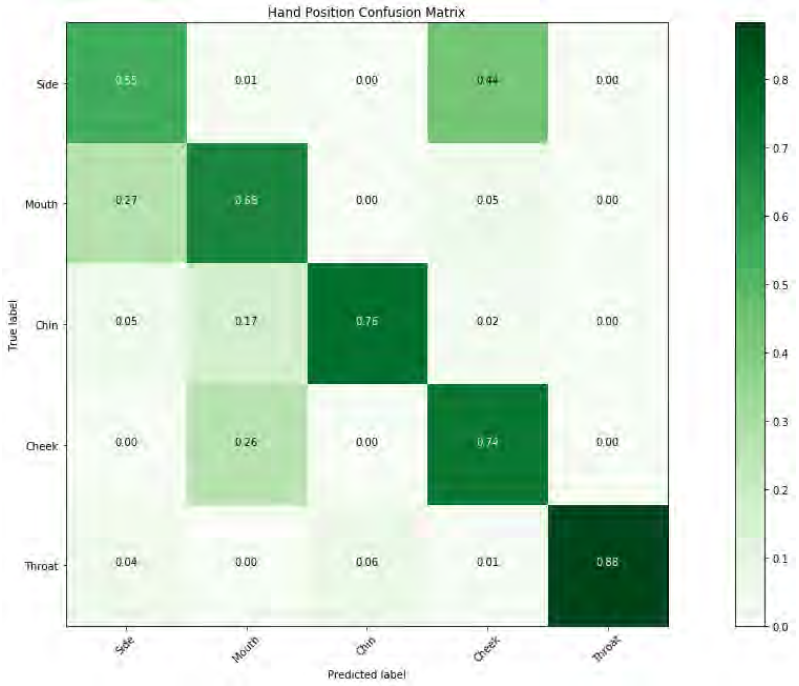
Σε αυτό το κεφάλαιο θα αναλυθούν για κάθε νευρωνικό δίκτυο τα αποτελέσματα εκπαίδευσή τους, με την βοήθεια μητρώων σύγχυσης. Αρχικά, τα δεδομένα μας στο σύνολο εκπαίδευσης και ελέγχου ήταν διαχωρισμένα σε 80-20. Η απόδοση κάθε νευρωνικού ως προς το σύνολο ελέγχου του παρουσιάζεται στον παρακάτω συγκεντρωτικό πίνακα.

	Hand Position NN	Manhattan Distance	Hand Shape CNN	Lips CNN
Accuracy	30%	75%	80%	45%

Πίνακας 5.1: Πίνακας απόδοσης κάθε νευρωνικού δικτύου

5.1 Αποτελέσματα εύρεσης θέσης χεριών

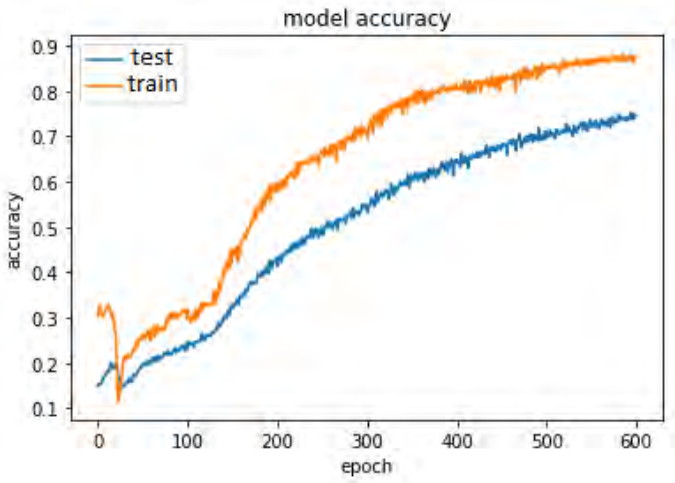
Η πρόβλεψη της θέσης των χεριών γίνεται μέσω ενός αλγορίθμου, ο οποίος κατηγοριοποιεί τις συντεταγμένες ανάλογα με την ελάχιστη απόσταση Manhattan από τα 5 κέντρα των κλάσεων. Όπως φαίνεται και στο Σχήμα 5.1 η πλειοψηφία των δεδομένων του σετ ελέγχου ταξινομείται επιτυχώς στις αντίστοιχες κλάσεις. Ωστόσο, παρατηρείται αυξημένη σύγχυση μεταξύ των κλάσεων “Side” - “Cheek”, ενώ μεταξύ “Side” - “Mouth” και “Mouth” - “Cheek” η διακριτότητα των κλάσεων είναι πολύ καλύτερη. Το γεγονός αυτό οφείλεται στο ότι οι αποστάσεις των κέντρων των κλάσεων που αντιπροσωπεύονται από συντεταγμένες εικονοστοιχείων είναι πολύ κοντινές.



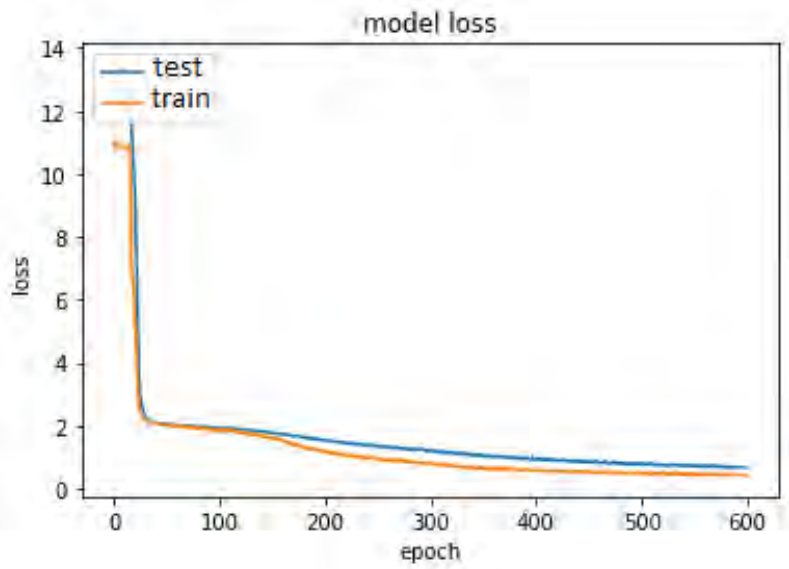
Σχήμα 5.1: Κανονικοποιημένο μητρώο σύγχυσης αλγορίθμου εύρεσης θέσης χεριού

5.2 Αποτελέσματα εύρεσης χειρονομιών

Όπως παρατηρείται στο Σχήμα 5.2 στο τέλος κάθε εποχής το μοντέλο αξιολογήθηκε με βάση τα σετ εκπαίδευσης και επικύρωσης, προκειμένου να παρατηρηθεί η εμφάνιση υπερβολικής προσαρμογής κατά τη διάρκεια της εκπαίδευσης. Ο στόχος είναι να σταματήσει η διαδικασία εκπαίδευσης νωρίτερα σε ένα σημείο όπου το μοντέλο δείχνει υψηλή ακρίβεια στα δεδομένα του σετ επικύρωσης, αλλά πριν υπερκεράσει σε μεγάλο βαθμό το σετ εκπαίδευσης. Ως εκ τούτου, σταματήσαμε την εκπαίδευση στο τέλος της 600ης εποχής για να αξιολογήσουμε το μοντέλο στο σετ δοκιμών.

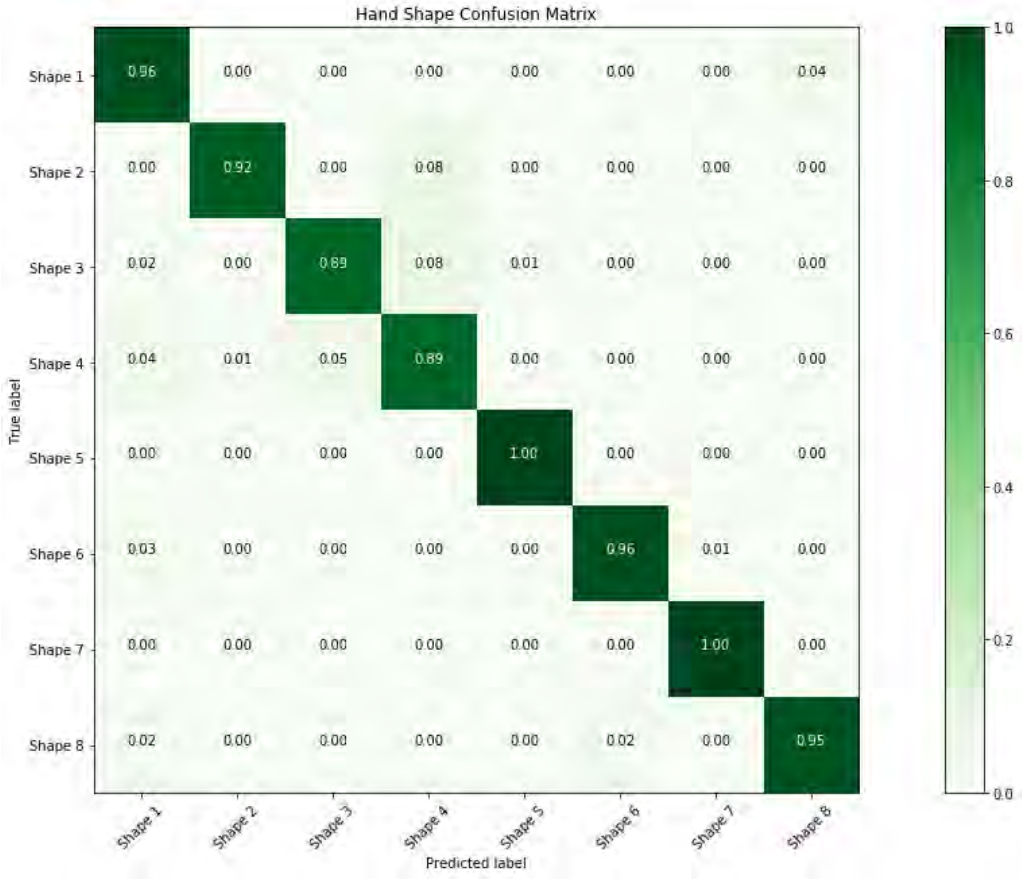


Σχήμα 5.2: Διάγραμμα συνάρτησης απόδοσης - χρόνου επαναλήψεων για το CNN χειρονομιών



Σχήμα 5.3: Διάγραμμα συνάρτησης σφάλματος - χρόνου επαναλήψεων για το CNN χειρονομιών

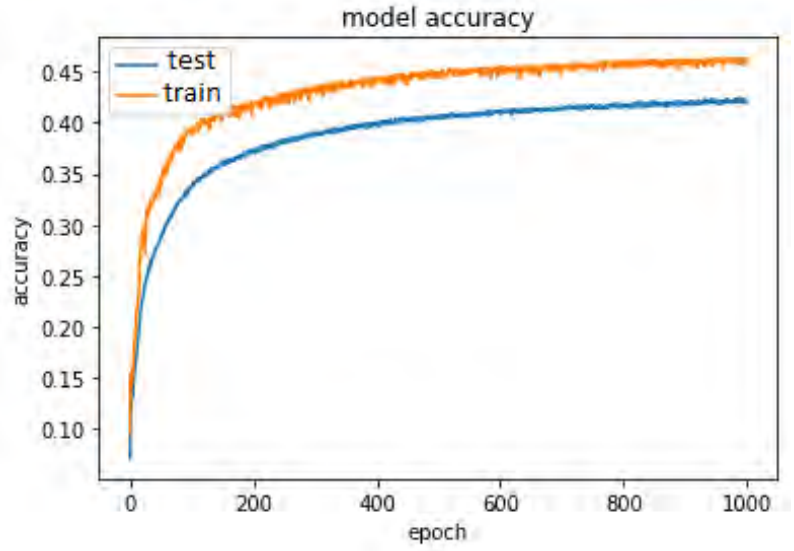
Παρατηρούμε από το Σχήμα 5.4 ότι για το συνελικτικό δίκτυο των χειρονομιών η πλειοψηφία των κλάσεων ταξινομείται αποτελεσματικά.



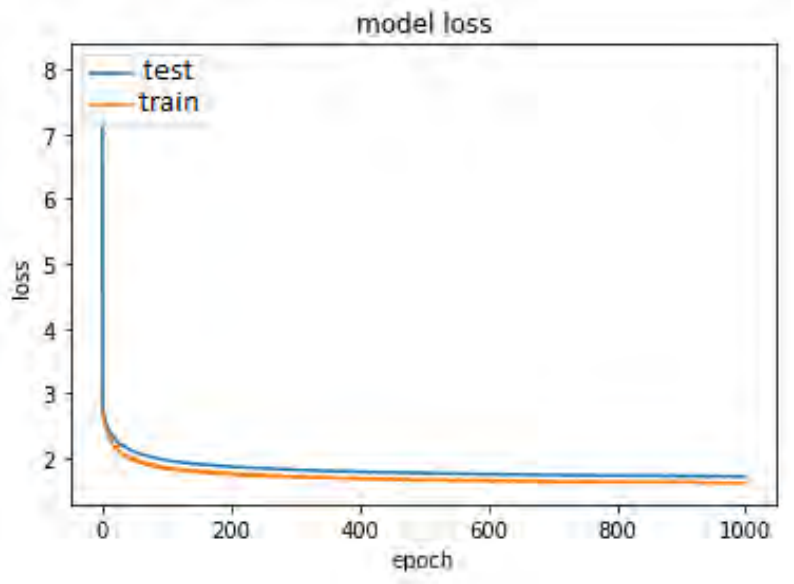
Σχήμα 5.4: Κανονικοποιημένο μητρώο σύγχυσης για το νευρωνικό δίκτυο των χειρονομιών

5.3 Αποτελέσματα εύρεσης σχήματος χειλιών

Αντίστοιχα διαγράμματα παρουσιάζονται κατά την εκπαίδευση του CNN για το σχήμα των χειλιών. Η εκπαίδευση σταματά μετά το πέρας της 1000στής εποχής.



Σχήμα 5.5: Διάγραμμα συνάρτησης απόδοσης-χρόνου επαναλήψεων για το CNN των χειλιών



Σχήμα 5.6: Διάγραμμα συνάρτησης σφάλματος-χρόνου επαναλήψεων των χειλιών

Για τις κλάσεις των χειλιών, η εκπαίδευση είναι αρκετά πιο περίπλοκη, καθώς οι διαφορές μεταξύ της προφοράς των φωνημάτων στα χείλη είναι δύσκολα διακριτές ακόμα και από τον ανθρώπινο μάτι. Έτσι, όπως παρατηρούμε και από το μητρώο σύγχυσης το ποσοστό ορισμένων κλάσεων για παράδειγμα της πέμπτης, έκτης και έβδομης δεν ταξινομείται σχεδόν ποτέ σωστά. Ωστόσο, για τις υπόλοιπες ένα ποσοστό της τάξης του 40-50 % κατηγοριοποιείται επιτυχώς.

Σχήμα 5.7: Κανονικοποιημένο μητρώο σύγχυσης για το νευρωνικό δίκτυο των χειλιών

5.4 Τελική Πρόβλεψη Φωνημάτων

Τα αποτελέσματα από το πρώτο απλό νευρωνικό είναι αναξιόπιστα. Η δεύτερη προσέγγιση, όμως αποτελεί μια εύκολη και αποτελεσματική λύση στην εύρεση της θέσης του χεριού γύρω από το πρόσωπο, ώστε να αποκωδικοποιηθεί το φωνήεν. Στην περίπτωση του συνελικτικού δικτύου για τις χειρονομίες, οι φωτογραφίες προβλέπονται αξιόπιστα σε κάθε μια από τις οκτώ κλάσεις.Το τελευταίο συνελικτικό δίκτυο αναφέρεται στην κατηγοριοποίηση του σχήματος των χειλιών, ένα απαιτητικό πρόβλημα όπου τα ποσοστά που λαμβάνει είναι ικανοποιητικά.

Αφού πλέον, έχουμε και τις 3 προβλέψεις που χρειαζόμαστε με μια απλή υλοποίηση μιας δομής if-elif-else μπορούμε να αντιληφθούμε ποιο φώνημα παρουσιάζεται. Για την αποκωδικοποίηση των συμφώνων, στην σιωπηρή ομιλία έχουμε 20 συνδυασμούς του τύπου της χειρονομίας με το σχήμα των χειλιών. Το ποσοστό επιτυχίας της πρόβλεψης ενός συμφώνου είναι 40.5%. Για την αποκωδικοποίηση των φωνηέντων έχουμε 14 συνδυασμούς της θέσης του χεριού γύρω από το πρόσωπο και του σχήματος των χειλιών, με ποσοστό επιτυχίας 33.75 %. Για τη συνολική εκτίμηση του ποσοστού επιτυχίας πρόβλεψης φωνημάτων θα πρέπει να λάβουμε υπόψιν την συχνό-

τητα εμφάνισης των φωνημάτων στην γαλλική γλώσσα. Τα ποσοστά εμφάνισης για τα σύμφωνα και τα φωνήεντα είναι 53.75% και 46.25% αντίστοιχα [17]. Επομένως, η υλοποίησή μας λαμβάνει ένα ποσοστό της τάξης του 37.4% για την συνολική πρόβλεψη φωνημάτων. Παρακάτω δίνεται ο ψευδοκώδικας για 7 από τους 34 συνολικούς συνδυασμούς των κλάσεων για την πρόβλεψη ενός φωνήματος.

Algorithm 1 Συνδυασμός προβλέψεων για τελική απόφαση φωνήματος

Result: The result is the predicted phoneme

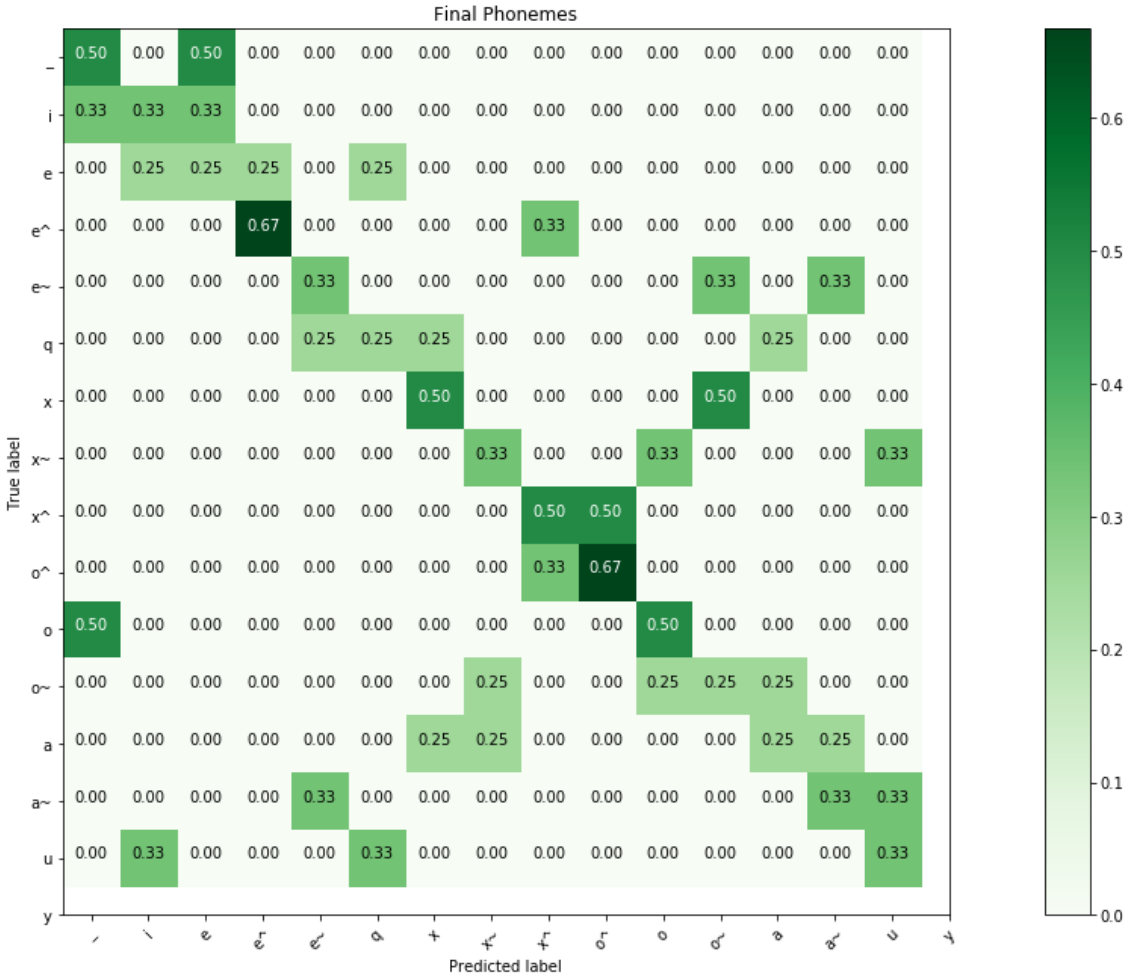
predictor_lips = model_Lips.predict(lip_photo)

predictor_handshape = model_Hand.predict(hand_photo)

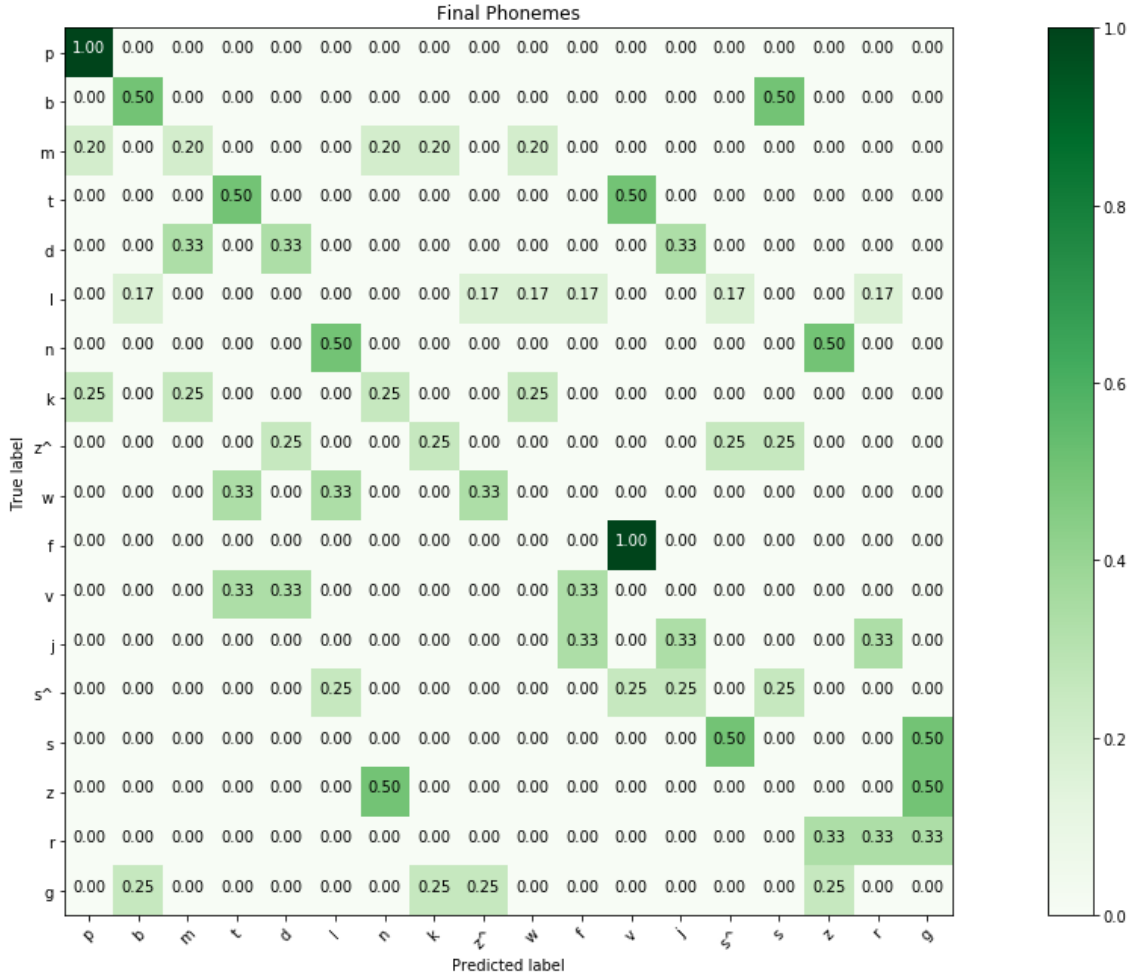
predictor_Manhattan = min_distance(center)

if *predictor_lips == 1 and predictor_handshape == 3* **then**
  print(”The predict phoneme is b”);
end
if *predictor_lips == 1 and predictor_handshape == 5* **then**
  print(”The predict phoneme is m”);
end
if *predictor_lips == 2 and predictor_handshape == 4* **then**
  print(”The predict phoneme is t”);
end
if *predictor_lips == 1 and predictor_handshape == 0* **then**
  print(”The predict phoneme is p”);
end
⋮
if *predictor_lips == 9 and predictor_Manhattan ==1* **then**
  print(”The predict phoneme is i”);
end
if *predictor_lips == 10 and predictor_Manhattan == 4* **then**
  print(”The predict phoneme is e”);
end
if *predictor_lips == 12 and predictor_Manhattan == 2* **then**
  print(”The predict phoneme is o”);
end

Οι Πίνακες 5.8 και 5.9 παρουσιάζουν τα μητρώα σύγχυσης των συνολικών κλάσεων των φωνημάτων. Λόγω του μεγάλου αριθμού των κλάσεων (34) διαχωρίζονται σε δυο μέρη, ώστε να παραμείνουν οι πίνακες ευανάγνωστοι. Οι υπολογισμοί των τελικών μητρώων σύγχυσης έγιναν με 40 φωτογραφίες ως το σύνολο δεδομένων, λόγω έλλειψης υπολογιστικής ισχύος.



Σχήμα 5.8: Κανονικοποιημένο μητρώο σύγχυσης για το συνδυαστικό σύστημα (Μέρος 1)



Σχήμα 5.9: Κανονικοποιημένο μητρώο σύγχυσης για το συνδυαστικό σύστημα (Μέρος 2)

5.5 Συζήτηση

Όλες οι εφαρμογές που ερευνήθηκαν σε αυτή την διπλωματική εργασία είναι video frames από τα δείγματα εκπαίδευσης και τα μοντέλα παράγουν προβλέψεις ανά frame, με βάση αρχιτεκτονικές 2D συνελικτικών δικτύων. Παρ' όλα αυτά, δεν διαμορφώνουν τη δυναμική των βίντεο και αποφεύγουν τις χρονικές πληροφορίες που μπορεί να βοηθήσουν στην παραγωγή ακριβέστερων προβλέψεων. Ένα 3D CNN, στο οποίο κάθε δείγμα εκπαίδευσης είναι ένα βίντεο και το μοντέλο δίνει τις προβλέψεις ανά βίντεο ήταν μια πιθανή προσέγγιση για την αναγνώριση λέξεων. Ωστόσο, αυτή η λύση είναι υπολογιστικά δαπανηρή με απαγορευτικά υψηλές απαιτήσεις μνήμης.

Κεφάλαιο 6

Μελλοντική Εργασία

Η παρούσα διπλωματική εργασία ασχολείται με την πρόβλεψη φωνημάτων ανά φωτογραφία που εξάγονται από βίντεο. Με την χρήση δύο συνελικτικών νευρωνικών δικτύων και ένα αλγόριθμο εύρεσης θέσης χεριού που βασίζεται στην Manhattan απόσταση επιτυγχάνεται η παραπάνω υλοποίηση. Ωστόσο, μια μελλοντική προσέγγιση θα μπορούσε να είναι η προσπάθεια αύξησης του ποσοστού επιτυχίας του νευρωνικού δικτύου για το σχήμα των χειλιών, καθώς επί της ουσίας συμβάλλει άμεσα στην κατανόηση για το αν το φώνημα που προβλέπεται είναι σύμφωνο ή φωνήεν. Μια επιπρόσθετη μελλοντική υλοποίηση θα μπορούσε να είναι η πρόβλεψη λέξεων από το συνολικό βίντεο, μια 3D δηλαδή υλοποίηση, που θα λαμβάνει υπόψιν και τις χρονικές πληροφορίες από το βίντεο.

Βιβλιογραφία

[1] Hidden Markov model toolkit. http://htk.eng.cam.ac.uk/.

[2] Kaldi. https://kaldi-asr.org/doc/about.html.

[3] Noureddine Aboutabit. *"Reconnaissance de la Langue Francaise Parlee Completee (LPC) : Decodage phonetique des gestes main-l'evres"*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, April 2007.

[4] Apoorva Agrawal. "Loss Functions and Optimization Algorithms. Demystified". *Medium*, September 2017.

[5] Anup Bhande. "What is underfitting and overfitting in machine learning and how to deal with it". *Medium*, March 2018.

[6] Jason Brownlee. "Why One-Hot Encode Data in Machine Learning?" *Machine Learning Mastery*, July 2017.

[7] Ting-Hao Chen. "What is padding in Convolutional Neural Network?" *Medium*, September 2017.

[8] Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. "Reducing Overfitting in Deep Networks by Decorrelating Representations". November 2016. https://arxiv.org/abs/1511.06068v4.

[9] Daphne Cornelisse. "An intuitive guide to Convolutional Neural Networks". *FreeCodeCamp*, April 2018.

[10] Victor Dibia. "How to Build a Real-time Hand-Detector using Neural Networks (SSD) on Tensorflow". *Medium*, December 2017.

[11] Rohith Gandhi. "A Look at Gradient Descent and RMSprop Optimizers". *Towards Data Science*, June 2018.

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *"Deep Learning"*. The MIT Press, 2016. http://www.deeplearningbook.org.

[13] Amazon Polly Developer Guide. "Phoneme and Viseme Table for French Language". 2018.

[14] Dishashree Gupta. "Fundamentals of Deep Learning – Activation Functions and When to Use Them?" *Analytics Vidhya*, October 2017.

[15] Martin T. Hagan, Howard B. Demuth, Mark H. Beale, and Orlando De Jesús. *"Neural Network Design"*. 2 edition, 1996.

[16] Nameer Hirschkind, Saruque Mollick, and Jyo Pari. "Convolutional Neural Network". *Brilliant Math and Science Wiki*, May 2019.

[17] Xah Lee. "French Letter Frequency". February 2019.

[18] Li Liu, Thomas Hueber, Gang Feng, and Denis Beautemps. *"Visual recognition of continuous Cued Speech using a tandem CNN-HMM approach"*. PhD thesis, Institute of Engineering Univ. Grenoble Alpes, Grenoble, France, December 2015.

[19] Li Liu, Thomas Hueber, Gang Feng, and Denis Beautemps. "Multimodal database of French Cued Speech". *Zenodo*, March 2018.

[20] Michael A. Nielsen. *"Neural Networks and Deep Learning"*, chapter "Improving the way neural networks learn". Determination Press, 2015.

[21] Paul-Louis Pröve. "An Introduction to different Types of Convolutions in Deep Learning". *Towards Data Science*, July 2017.

[22] Russell Reed and Robert J MarksII. *"Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks"*. The MIT Press, London, 5 edition, 1999.

[23] Adrian Rosebrock. "A simple neural network with Python and Keras". *Pyimagesearch*, September 2016.

[24] Adrian Rosebrock. "Facial landmarks with dlib, OpenCV, and Python". *Pyimagesearch*, April 2017.

[25] Rajalingappaa Shanmugamani. *"Deep Learning for Computer Vision"*. Packt Publishing Ltd, Birmingham, 1 edition, 2018.

[26] L. Steinberg. "What is Cued Speech". *Cued Speech UK*, 2015. `https://www.cuedspeech.co.uk/`.

[27] S. Theodoridis and K.Koutroumbas. *Pattern Recognition*, pages 509–510. Elsevier Inc, 4 edition, 2012.

[28] Muneeb ul Hassan. "Convolutional Network for Classification and Detection". *Neurohive*, November 2018.

[29] Shiva Verma. "Understanding different Loss Functions for Neural Networks". *Towards Data Science*, June 2018.

[30] Anish Singh Walia. “Types of Optimization Algorithms used in Neural Networks and Ways to Optimize Gradient Descent”. *Towards Data Science*, June 2017.

[31] Matthew D. Zeiler and Rob Fergus. “Visualizing and understanding convolutional networks”. In *Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV*, 2014.

Συντομογραφίες

κ.λπ.	και λοιπά
κ.ο.κ	και ούτω καθεξής
π.χ.	παραδείγματος χάριν
CNN	Convolutional Neural Network
ReLU	Recticied Linear Unit
CS	Cued Speech
Adam	Adaptive Moment Estimation
HTK	Hidden Markov ModelToolkit
MSE	Mean Square Error
fps	frames per second

Ορολογία - Γλωσσάρι

Ελληνικός όρος	Αγγλικός όρος
εσσιανός πίνακας	hessian matrix
μήτρα σύγχυσης	confusion matrix
συνελικτικό δίκτυο	convolutional network
σιωπηρή ομιλία	cued speech
Μέθοδος απότομης καθόδου	Gradient Descent
παρτίδα	batch
Υπερπροσαρμογή	Overfitting
χάρτης χαρακτηριστικών	feature map
Επόχη	Epoch
εικονοστοιχεία	pixels
σιγμοειδής	sigmoid
βελτιστοποιητές	optimizers
συνάρτηση κόστους	loss function
υποπροσαρμογή	underfitting
σε κλίμακα του γκρι	grayscaled
συγκεντρωτικό στρώμα	pooling layer
Υπέρ-παράμετροι	Hyper-parameters
στόχος	target