ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Μελέτη μεθόδων μετάφρασης νοηματικής γλώσσας από βίντεο»

Του φοιτητή
Δάρα Ιωάννη
Αρ. Μητρώου: 2019033

Επιβλέπων
Μπράτσας Χαράλαμπος
Επίκουρος Καθηγητής

26 Ιανουαρίου 2025

Τίτλος Δ.Ε. Μελέτη μεθόδων μετάφρασης νοηματικής γλώσσας από βίντεο
Κωδικός Δ.Ε. 24238
Ονοματεπώνυμο φοιτητή/ών Δάρας Ιωάννης
Ονοματεπώνυμο εισηγητή Μπράτσας Χαράλαμπος
Ημερομηνία ανάληψης Δ.Ε. 01/10/2024
Ημερομηνία περάτωσης Δ.Ε. 26/01/2025

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Σε όσους με υποστήριξαν και έκαναν δυνατή την ολοκλήρωση αυτής της εργασίας»

Πρόλογος

Το θέμα επιλέχθηκε με γνώμονα δύο βασικούς λόγους. Ο πρώτος, ήταν η αναζήτηση ενός αντικειμένου, η μελέτη του οποίου θα μπορούσε να έχει ουσιαστικό αντίκτυπο και να συνεισφέρει στον τομέα του. Ο δεύτερος λόγος, ήταν το μεγάλο προσωπικό ενδιαφέρον για τον τομέα των νευρωνικών δικτύων, τόσο για τις καινοτόμες ιδέες που αντιπροσωπεύουν, όσο και για τα πολλαπλά οφέλη που μπορούν να προσφέρουν. Συνδυάζοντας αυτούς τους δύο λόγους, από τις διαθέσιμες επιλογές, επιλέχθηκε η μελέτη μεθόδων μετάφρασης νοηματικής με την ελπίδα πως η εργασία που θα πραγματοποιηθεί θα αποτελέσει ένα ίσως μικρό, αλλά ουσιαστικό βήμα προς τη μείωση των δυσκολιών που αντιμετωπίζουν τα άτομα με προβλήματα ακοής και ομιλίας. Η εκπόνηση αυτής της διπλωματικής οδήγησε στην βαθύτερη κατανόηση της δομής και λειτουργίας των νευρωνικών δικτύων, με έμφαση στα συνελικτικά, καθώς και της δομής και μορφολογίας της νοηματικής γλώσσας και των δυσκολιών που αντιμετωπίζονται για την αυτόματη μετάφραση της.

Περίληψη

Η νοηματική γλώσσα περιλαμβάνει την κίνηση των χεριών, εκφράσεις του προσώπου αλλά και όλου του σώματος για να μεταφέρει πληροφορία και αποτελεί την πιο διαδεδομένη μέθοδο επικοινωνίας για άτομα με προβλήματα ακοής και ομιλίας. Αυτά τα χαρακτηριστικά την καθιστούν ιδιαίτερα προκλητική για τον σκοπό την αυτόματης μετάφρασης με τεχνολογικά μέσα. Τα νευρωνικά δίκτυα έχουν αποδειχθεί ιδιαίτερα αποτελεσματικά στην αντιμετώπιση αυτού του προβλήματος, με νέες μεθοδολογίες και μοντέλα να διαπρέπουν ολοένα και περισσότερο στην αναγνώριση και μετάφραση νοηματικής γλώσσας, αξιοποιώντας διάφορα δημόσια διαθέσιμα σύνολα δεδομένων. Αυτή η εργασία έχει ως σκοπό να μελετήσει τέτοιες μεθόδους, με τα σύνολα δεδομένων που χρησιμοποιούνται και να προσπαθήσει να κάνει δύο βασικές συνεισφορές. Η πρώτη είναι η τροποποίηση ενός μοντέλου προκειμένου να πετύχει την μείωση των υπολογιστικών απαιτήσεων, της επιτάχυνσής του και την δυνατότητα εκπαίδευσης του σε μη εξειδικευμένο υλικό. Οι παραλλαγές περιλαμβάνουν την χρήση διαφορετικών συνελικτικών δικτύων (ResNet18, SqueezeNet, ShuffleNetv2) και τη μείωση του μεγέθους των εικόνων στην είσοδο, χρησιμοποιώντας περικοπή. Με αυτές τις απλές μεθόδους, εξετάστηκαν τα θετικά και τα αρνητικά σε κάθε περίσταση, με ένα από τα καλύτερα αποτελέσματα, να περιλαμβάνει τον διπλασιασμό της ταχύτητας και τον υποδιπλασιασμό των υπολογιστικών αναγκών, με λιγότερο από 8% μείωση στην απόδοση (ResNet18 με 150×150 εισόδους). Επιπλέον, παρατηρήθηκε ένα ενδιαφέρον φαινόμενο: η απόδοση δεν μειωνόταν αναλογικά με το μέγεθος των εισόδων, αλλά παρουσίασε μια αρχική μείωση και στη συνέχεια έμεινε περίπου σταθερή μέσα σε ένα συγκεκριμένο εύρος εισόδων. Η δεύτερη, είναι η απόπειρα συνεισφοράς στον τομέα της ελληνικής νοηματικής, προωθώντας και αξιολογώντας το Greek Sign Language Dataset, διασταυρώνοντας παράλληλα την δυνατότητα γενίκευσης των υπόλοιπων αποτελεσμάτων.

«A study of sign language translation methods from video»

«Ioannis Daras»

Abstract

Sign language uses hand movements, along with with face expressions and body movements to convey information and it is the most widespread form of communication for people with hearing and speaking problems. Those characteristics make the purpose of automatic translation via technological means, a challenge. Neural networks have proven highly effective in addressing this problem, with new methodologies and models increasingly excelling at recognizing and translating sign languages, leveraging several publicly available datasets. The goal of this thesis is to study such methods, alongside commonly used datasets, and to contribute in two key ways. The first, is choosing and altering a model for the purpose of reducing its computational needs and its speed, as well as making it trainable in non-specialized hardware setups. The alterations include using different convolutional networks (ResNet18, SqueezeNet, ShuffleNetv2) and reducing the input size of the images by cropping them. Using these simple methods, several trade-offs were examined, with one of the best being a doubling of speed and a halving of computational requirements, while accuracy decreased by less than 8% (ResNet18 with 150×150 inputs). Furthermore, an interesting phenomenon was observed: accuracy did not decline proportionally to the input reduction but instead experienced an initial drop and then stabilized within a specific range of inputs. The second contribution, is one specifically in the field of Greek sign language and it includes the attempt to review and evaluate the Greek Sign Language Dataset, for its real-world application capabilities, while at the same time testing the generalizability of the rest of the results provided.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή της διπλωματικής μου εργασίας, τον καθηγητή κ. Μπράτσα Χαράλαμπο για την καθοδήγηση και τον χρόνο που αφιέρωσε. Θα ήθελα επίσης να ευχαριστήσω τον υποψήφιο διδάκτορα κ. Ιωαννίδη Λάζαρο για την πολύτιμη βοήθεια και τις συμβουλές που παρείχε καθ' όλη τη διάρκεια εκπόνησης. Επίσης, θα ήθελα να ευχαριστήσω όσους συμπαραστάθηκαν και συνέβαλλαν με την υποστήριξη τους, στην ολοκλήρωση αυτής της εργασίας.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Συντομογραφίες

Δ.Ε. Διπλωματική Εργασία
ΔΙΠΑΕ Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε. Πτυχιακή Εργασία
CNN Convolutional Neural Network
GSL-SI Greek Sign Language - Signer Independent
GSL-SD Greek Sign Language - Signer Dependent
WER Word Error Rate
CTC Connectionist Temporal Classification
BiLSTM Bidirectional Long Short-Term Memory
ReLU Leaky Rectified Linear Unit
GPU Graphics Processing Unit
FLOP Floating Point Operation
CSLR Continuous Sign Lanuage Recognition
SLT Sign Language Translation
ISLR Isolated Sign Language Recognition
HMM Hidden Markov Models
lr learning rate
VA Virtual Alignment VE Visual Enhancement

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Η νοηματικές γλώσσες αποτελούν τον πιο διαδεδομένο τρόπο επικοινωνίας για τις κοινότητες των ατόμων με προβλήματα ακοής και ομιλίας. Η επίσημη εμφάνιση της στην ιστορία χρονολογείται στα μέσα του 1600, αλλά στη πραγματικότητα οι άνθρωποι χρησιμοποιούν χειρονομίες για να επικοινωνήσουν πληροφορίες από πολύ νωρίτερα, πιθανώς ακόμα και πριν την ομιλία. Παρ ’όλα αυτά, η σημερινή δομή της μέσης κοινωνικής ζωής δεν προνοεί για άτομα με τέτοια προβλήματα και οι ενέργειες που γίνονται προς αυτή τη κατεύθυνση είναι πολύ μικρές. Η τεχνολογία, όμως, παρέχει πλήθος δυνατοτήτων σε όσους μπορούν να την αξιοποιήσουν κατάλληλα.

Μία από αυτές είναι η προσπάθεια να γεφυρώσει και το χάσμα της επικοινωνίας μεταξύ ατόμων με τέτοια προβλήματα και του υπόλοιπου κόσμου, μέσω της χρήσης των νευρωνικών δικτύων και της βαθιάς μάθησης. Αυτές οι τεχνολογίες έχουν γνωρίσει μεγάλη απήχηση τα τελευταία χρόνια λόγω των επιδόσεων τους σε μεγάλα προβλήματα και της εξέλιξης συσκευών όπως οι κάρτες γραφικών και χρησιμοποιήθηκαν για την εξέλιξη κλάδων όπως η αναγνώριση φυσικής γλώσσας και η αναγνώριση χειρονομιών από αισθητήρες. Μέσα σε αυτούς είναι και ο κλάδος της υπολογιστικής όρασης, ο οποίος άνθισε με την εμφάνιση των συνελικτικών δικτύων (Convolutional Neural Networks - CNNs). Ένας συνδυασμός από αυτούς τους κλάδους έδωσε τη σημερινή μορφή του τομέα μετάφρασης νοηματικής, ο οποίος αξιοποιεί τεχνικές που χρησιμοποιούν CNN και τεχνικές μετάφρασης γλώσσας προκειμένου να καταφέρει να μεταφράσει νοήματα καταγεγραμμένα σε βίντεο, σε γραπτή γλώσσα. Παράλληλα με τις τεχνολογίες, δημοσιεύτηκαν μεγάλα σύνολα δεδομένων, διαθέσιμα σε όλους, κατάλληλα δομημένα για την εκπαίδευση δικτύων με σκοπό την αναγνώριση και μετάφραση της νοηματικής, τα οποία βοήθησαν στην περαιτέρω εξέλιξη της έρευνας γύρω από αυτό το αντικείμενο.

1.2 Στόχος Διπλωματικής

Παρά τις προσπάθειες, ιδίως τα τελευταία χρόνια, για τη δημιουργία αξιόπιστων μοντέλων για την αναγνώριση και μετάφραση της νοηματικής, ακόμα τα αποτελέσματα δεν είναι αρκετά ικανοποιητικά, τόσο στο κομμάτι της απόδοσης τους, όσο και στο κομμάτι της αδυναμίας τους να αντεπεξέλθουν σε πραγματικές συνθήκες. Τα περισσότερα μοντέλα που δοκιμάζονται δίνουν έμφαση στη βελτίωση της απόδοσης, ενώ απαιτούν πολύ κοστοβόρο υλικό για να εκπαιδευτούν. Πάνω σε αυτό το πρόβλημα προσπαθεί να προσπαθεί να προσφέρει καινούριες πληροφορίες η παρούσα εργασία, δοκιμάζοντας πιο ”ελαφριές” παραλλαγές μοντέλου, που μπορούν να εκπαιδευτούν χρησιμοποιώντας πολύ υποδεέστερα μηχανήματα. Σκοπός είναι τόσο να μελετηθούν τα αρνητικά και τα θετικά τέτοιων παραλλαγών, όσο και να επιτρέψει και να προωθήσει την ερευνητική δουλειά πάνω σε αυτόν τον τομέα, αφαιρώντας το εμπόδιο της ανάγκης για ακριβό και εξειδικευμένο υλικό. Παράλληλα, γίνεται μια μικρή αξιολόγηση ενός συνόλου δεδομένων που αφορά την ελληνική νοηματική γλώσσα, με σκοπό τον έλεγχο της δυνατότητας αξιοποίησής του σε πραγματικές συνθήκες, αλλά και τη διασταύρωση των αρχικών αποτελεσμάτων από το πρώτο σύνολο δεδομένων, το οποίο χρησιμοποιείται ως σημείο αναφοράς στη βιβλιογραφία.

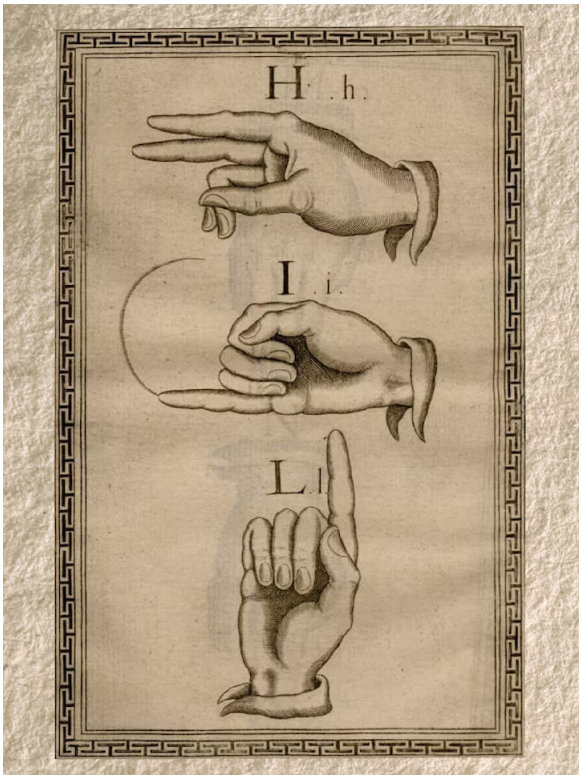
1.3 Δομή Διπλωματικής

Η δομή έχει οριστεί με γνώμονα την αρχική εισαγωγή του αναγνώστη στο πεδίο της νοηματικής γλώσσας, της έρευνας που έχει γίνει στον τεχνολογικό τομέα σχετικά με αυτήν και στη συνέχεια την κατανόηση των βημάτων που ακολουθήθηκαν για τη πραγματοποίηση των πειραμάτων και στα αποτελέσματα που προέκυψαν. Συγκεκριμένα, στο δεύτερο κεφάλαιο ακολουθεί ιστορική ανασκόπηση της νοηματικής γλώσσας και παρουσίαση των δομικών της στοιχείων. Στο τρίτο κεφάλαιο γίνεται μια βιβλιογραφική ανασκόπηση από σημαντικές δημοσιεύσεις πάνω στον τομέα της μετάφρασης νοηματικής γλώσσας και της ιστορίας του. Το τρίτο κεφάλαιο περιλαμβάνει βασικά θεωρητικά στοιχεία απαραίτητα για την κατανόηση του μοντέλου που χρησιμοποιείται στο τέταρτο κεφάλαιο, το οποίο περιγράφει τη μεθοδολογία που ακολουθήθηκε για να αντιμετωπιστεί το πρόβλημα, καθώς και τα βήματα που ακολουθήθηκαν για την εξαγωγή των αποτελεσμάτων. Τα αποτελέσματα αυτά παρουσιάζονται στο πέμπτο κεφάλαιο, ενώ στο έκτο, ακολουθεί ο σχολιασμός τους και συζήτηση σχετικά με τις δυσκολίες που παρουσιάστηκαν κατά την εκπόνηση. Το τελευταίο κεφάλαιο είναι μία σύνοψη της δουλειάς που έγινε και ορισμένες προτάσεις για επόμενα βήματα που θα μπορούσαν να γίνουν.

Κεφάλαιο 2ο: Ανασκόπηση νοηματικής γλώσσας και ιστορικό υπόβαθρο

2.1 Ιστορικό υπόβαθρο

Η ιστορία της νοηματικής γλώσσας είναι πολύ πιθανόν να υπάρχει όσο υπάρχουν και ανθρώπινοι πολιτισμοί. Σε όλη τη διάρκεια της ιστορίας, υπάρχουν δείγματα πως οι άνθρωποι χρησιμοποιούσαν διάφορους τρόπους επικοινωνίας, χωρίς να περιορίζονται μόνο στην λεκτική. Ένα από τα πρώτα καταγεγραμμένα παραδείγματα αποτελεί ο μοναχός Ponce de León, ο οποίος χρησιμοποιούσε χειρονομίες στο μοναστήρι του προκειμένου να διδάξει έναν τρόπο επικοινωνίας σε όσους είχαν απώλεια ακοής. Επηρεασμένος από την δουλεία του de León, ο κληρικός και γλωσσολόγος Juan Pablo Bonet δημοσίευσε το 1620, την πρώτη καταγεγραμμένη δουλειά με σκοπό την εκπαίδευση ανθρώπων με απώλεια ακοής. Σε αυτή παρουσιάζεται ένα αναλυτικό αλφάβητο το οποίο αντιστοιχούσε κάθε γράμμα σε μία χειρονομία του δεξιού χεριού με σκοπό την σύνθεση λέξεων, γράμμα προς γράμμα. [1].



Σχήμα 2.1: [1] Χειρονομίες-γράμματα από το λεξικό του Juan Pablo Bonet (1620)

Μέχρι και αυτή την περίοδο, είναι εμφανές πως ενώ υπάρχουν είδη επικοινωνίας που αξιοποιούν τη χρήση χειρονομιών, δεν υπάρχει μια γλώσσα που μπορεί να αξιοποιηθεί για καθημερινή χρήση όπως μία λεκτική γλώσσα. Αυτός είναι και ο λόγος που η επίσημη ιστορία της νοηματικής, με τη μορφή που είναι γνωστή σήμερα, ξεκινάει το 1750 στη Γαλλία, με την απόφαση του Γάλλου ιερέα Abbé de l'Epée, να ξεκινήσει να τη διδάσκει αφιλοκερδώς, σε έναν μικρό αριθμό ατόμων [2]. Συνειδητοποιώντας την απήχηση της δουλείας του και τη βοήθεια που προσέφερε, ίδρυσε το Institut National des Jeunes Sourds de Paris (Εθνικό Ίδρυμα για παιδιά με απώλεια ακοής του Παρισιού), το πρώτο δημόσιο ίδρυμα με σκοπό την εκπαίδευση ατόμων με προβλήματα ακοής. Εκεί εδραίωσε την πρώιμη μορφή της νοηματικής, χρησιμοποιώντας για πρώτη φορά ένα σύστημα αρκετά σύνθετο ώστε να επιτρέπει την έκφραση γραμματικών

και συντακτικών δομών παρόμοιων με εκείνες της κοινής ομιλίας. Συγκεκριμένα, έθεσε τα θεμέλια για μια διάλεκτο η οποία δεν αποτελείται μόνο από μία αντιστοιχία λέξεων-νοημάτων αλλά περιλαμβάνει φωνολογική, μορφολογική και σημειολογική οργάνωση.

2.2 Νοηματική Γλώσσα

2.2.1 Δομικά Στοιχεία

Το συντακτικό σύστημα της νοηματικής αποτελείται από μία δομική μονάδα, το μόρφημα, το οποίο όμως, λόγω της φύσης της νοηματικής, μπορεί να προέλθει από δύο διαφορετικά συντακτικά συστήματα. Ως μόρφημα ορίζεται το μικρότερο κομμάτι γλώσσας με δική του σημασία, το οποίο είναι μέρος μίας λέξης ή μία ολόκληρη λέξη. Για παράδειγμα, η αγγλική λέξη 'Worker', περιλαμβάνει τα μορφήματα "work" και "-er". Έτσι στη νοηματική, μία λέξη μπορεί να αναλυθεί στα γράμματα της και να συλλαβιστεί με μία σειρά νοημάτων, καθένα από τα οποία αντιστοιχίζεται άμεσα με ένα συγκεκριμένο γράμμα της αλφαβήτου (finger-spelling), αλλά παράλληλα και ολόκληρη η λέξη μπορεί να αποτελεί ένα μόρφημα ή ένα συνδυασμό μορφημάτων. Με αυτόν τον τρόπο, μπορούν να χωριστούν δύο διαφορετικοί τρόποι επικοινωνίας με τη χρήση νοηματικής, αυτός που κάνει χρήση συλλαβισμού με χειρονομίες και αυτός που χρησιμοποιεί χειρονομίες ως ολόκληρες λέξεις φράσεις. Οι συγγραφείς του [2] χρησιμοποιούν ως παράδειγμα τη φράση "Ναι, τον ξέρω", προκειμένου να δώσουν έμφαση στη διαφορά αυτών των δύο μεθόδων, η οποία μπορεί να έχει τη σημασία "Ναι, τον γνωρίζω", αλλά μπορεί επίσης να σημαίνει "Α ναι, αυτό περίμενα από αυτόν". Με την χρήση finger-spelling, ο μόνος τρόπος να γίνει διάκριση μεταξύ των δύο αυτών αποδόσεων, είναι μέσω εκφράσεων στο πρόσωπο του ομιλητή και για αυτό τον λόγο αυτός ο τρόπος θεωρείται πιο πολύ ένα είδος "συμβολικού συστήματος" παρά είδος νοηματικής γλώσσας. Αντίθετα, με τη χρήση χειρονομιών υπάρχει γλωσσικό περιθώριο λόγω της φύσης τους, και για αυτό συναντώνται πιο συχνά στη νοηματική γλώσσα που χρησιμοποιείται στη καθημερινότητα.

Μία χειρονομία αποτελείται από έναν συνδυασμό χαρακτηριστικών, τα οποία χωρίζονται στα χαρακτηριστικά χεριών (σχήμα χεριών και δακτύλων, τοποθέτηση, κίνηση, κατεύθυνση παλάμης/δακτύλων) και στα υπόλοιπα χαρακτηριστικά (βλέμμα, κίνηση κεφαλιού, κίνηση ώμων, εκφράσεις και χειρονομίες στόματος, ταχύτητα εκτέλεσης). Ένα βασικό στοιχείο που θα χρησιμοποιηθεί σε μεγάλο βαθμό σε αυτή την εργασία, είναι η γλωσσική μονάδα (gloss), η οποία είναι ένας συνδυασμός των παραπάνω χαρακτηριστικών και αποτελεί τον δομικό λίθο των νοηματικών γλωσσών, όπως η λέξεις αποτελούν τον δομικό λίθο των γραπτών γλωσσών. Κατά τη μετάφραση της νοηματικής σε γραπτή γλώσσα, η γλωσσική μονάδα είναι η πιο κοντινή ερμηνεία που μπορεί να δοθεί σε μία χειρονομία ή σε μία σειρά χειρονομιών.

πχ. Μία σειρά χειρονομιών θα μπορούσε να μεταφραστεί σε γλωσσικές μονάδες ως "ΠΡΟΒΛΗΜΑ ΤΙ" το οποίο στον γραπτό λόγο αντιστοιχεί στη φράση "ποιο είναι το πρόβλημα".

Ένα ακόμα στοιχείο που παρατηρείται κατά τη χρήση της νοηματικής γλώσσας, είναι το γραμματικό μέσο της "αλλαγής-ρόλου", στο οποίο ο ομιλητής μπορεί να χρησιμοποιήσει (δείχνοντας συνήθως) αντικείμενα από το περιβάλλον του προκειμένου να αντικαταστήσει τη χρήση προσωπικών αντωνυμιών (πχ. εγώ, αυτός -ή), οι οποίες δεν υπάρχουν. Τέλος, όπως και στον προφορικό λόγο, μπορεί να παρατηρηθεί διαφοροποίηση μεταξύ γλωσσικών αποδόσεων ανάλογα με προηγούμενες ή επόμενες λέξεις κατά τη διάρκεια μίας έκφρασης. Είναι φανερό λοιπόν, πως η νοηματικές γλώσσες στη σημερινή επο-

Σχήμα 2.2: Το ελληνικό αλφάβητο αποτυπωμένο μέσω χειρονομιών στη νοηματική γλώσσα.

χή έχουν σημαντικές διαφορές σε σχέση με την αυστηρή και τηλεγραφική μορφή που είχαν στην αρχή, αλλά αντιθέτως μοιράζονται κοινά χαρακτηριστικά με τις γραπτές γλώσσες. Αυτό αποτελεί ένα θετικό χαρακτηριστικό για την κοινότητα των ανθρώπων με προβλήματα ομιλίας ή ακοής, ωστόσο, καθιστούν τη μετάφρασης της νοηματικής γλώσσας σε γραπτό λόγο ένα ιδιαίτερα δύσκολο εγχείρημα.

2.2.2 Η σημασία της νοηματικής γλώσσας για την κοινότητα των κωφών

Στη σημερινή εποχή, σύμφωνα με την Παγκόσμια Ομοσπονδία Κωφών, υπάρχουν περίπου 70 εκατομμύρια άνθρωποι με προβλήματα ή ολική απώλεια ακοής [3] [4], ενώ παράλληλα υπάρχουν πάνω από 200 διαφορετικές νοηματικές γλώσσες που χρησιμοποιούνται ανά τον κόσμο. Οι νοηματικές γλώσσες παρέχουν σε αυτούς τους ανθρώπους έναν φυσικό τρόπο επικοινωνίας τόσο μεταξύ τους, όσο και με τον υπόλοιπο κόσμο, διευκολύνοντας την ομαλή ένταξη τους στη κοινωνία. Για πολλούς, η νοηματική δεν αποτελεί απλά μία μέθοδο επικοινωνίας, αλλά ένα σημαντικό στοιχείο πολιτισμικής ταυτότητας, η οποία συμβάλλει στη διατήρηση και επέκταση της ιστορίας και των παραδόσεών τους, όπως άλλωστε και κάθε άλλη γλώσσα. Επιπροσθέτως, αποτελεί ένα εργαλείο με το οποίο αυτοί οι άνθρωποι μπορούν να είναι αυτόνομοι σε έναν κόσμο στον οποίο το βασικό μέσο επικοινωνίας είναι η ομιλία.

2.2.3 Επικοινωνιακά φράγματα

Κάποια από τα προβλήματα που αντιμετωπίζει η κοινότητα των κωφών είναι:

1. Περιορισμένη πρόσβαση σε διερμηνείς: Οι πιστοποιημένοι διερμηνείς είναι περιορισμένοι και δεν είναι πάντα εφικτό να βρεθούν σε καταστάσεις όπως νοσοκομεία, νομικές διαδικασίες ή εκπαιδευ-

τικά ιδρύματα. Αυτό αναγκάζει τα άτομα με προβλήματα ακοής να καταφεύγουν σε μη επαρκείς τρόπους επικοινωνίας, καθιστώντας τους επιρρεπείς σε παρανοήσεις. Ιδιαίτερα σε τομείς που αφορούν την υγεία, η αδυναμία επικοινωνίας στον βαθμό που πρέπει μπορεί να προκαλέσει πληθώρα προβλημάτων.

2. Προκλήσεις στον εργασιακό χώρο: Η δυσκολία επικοινωνίας μπορεί να επηρεάσει τόσο τη δουλειά όσο και τη κοινωνική αποδοχή αυτών των ατόμων.

3. Κοινωνικός αποκλεισμός: Οι κοινωνικές αλληλεπιδράσεις συχνά αποτελούν εμπόδιο για πολλά άτομα, με ιδιαίτερο πρόβλημα τις ομαδικές συζητήσεις όπου καθίσταται πιο δύσκολες και τεχνικές όπως το διάβασμα χειλιών.

4. Στίγμα και διακρίσεις: Παρόλο που στη σύγχρονη κοινωνία γίνονται βήματα προς την εξάλειψη φαινομένων προκατάληψης και στερεοτύπων, υπάρχουν ακόμα περιθώρια βελτίωσης, μέχρι να υπάρχει ίση αντιμετώπιση αυτών των ανθρώπων.

   Μία λύση για πολλά από αυτά τα προβλήματα θα ήταν η συστηματική εκμάθηση της νοηματικής γλώσσας, από τους πολίτες που δεν έχουν κάποιο πρόβλημα ακοής. Δυστυχώς, η συγκεκριμένη λύση αντιμετωπίζει ορισμένα προβλήματα. Η περιορισμένη έκθεση στη νοηματική γλώσσα, σε συνδυασμό με έλλειψη ευκαιριών για την πρακτική εφαρμογή της, αποτελούν σημαντικά εμπόδια. Επίσης η απουσία διδασκαλίας νοηματικής στα σχολεία, αυξάνει σημαντικά τη δυσκολία έναρξης της εκμάθησης. Αυτό οφείλεται τόσο στο οικονομικό κόστος που συνεπάγεται η πρόσληψη ιδιωτικού δασκάλου, όσο και στη δυσκολία εύρεσης κατάλληλου εκπαιδευτικού. Επιπλέον, η ουσιώδης διαφορά της νοηματικής από τις ομιλούμενες γλώσσες δημιουργεί πρόσθετες προκλήσεις για όσους επιθυμούν να την μάθουν. Ωστόσο, αξίζει να σημειωθεί πως τα τελευταία χρόνια έχουν αρχίσει να αναπτύσσονται κάποιες πλατφόρμες και να διανέμεται δωρεάν υλικό εκμάθησης, τα οποία εάν συνεχίσουν να βελτιώνονται, μπορούν να συμβάλλουν σημαντικά στην διάδοση της νοηματικής και στο κοινό που δεν έχει προβλήματα ακοής.

Κεφάλαιο 3ο: Βιβλιογραφική Ανασκόπηση

Όπως αναφέρθηκε παραπάνω, η νοηματική γλώσσα αποτελεί καίριο κομμάτι της ζωής των ανθρώπων με προβλήματα ακοής. Η αυτόματη μετάφρασης νοηματικής γλώσσας μπορεί να γεφυρώσει το επικοινωνιακό χάσμα που υπάρχει μεταξύ των δύο κόσμων και για αυτό το λόγο έχει αποτελέσει αντικείμενο εκτενούς έρευνας για πολλά χρόνια. Παρ'όλα αυτά η αυτόματη μετάφραση της νοηματικής αποτελεί ακόμα και σήμερα μία μεγάλη πρόκληση στους τομείς της υπολογιστικής όρασης και της επεξεργασίας φυσικής γλώσσας. Παρακάτω, υπάρχει μια σύντομη ανασκόπηση ορισμένων μεθοδολογιών που αξιοποιήθηκαν ανά τα χρόνια προς την επίτευξη αυτού του στόχου. Συγκεκριμένα, για έναν σαφή και λογικό διαχωρισμό των μεθόδων, υιοθετείται η πιο διαδεδομένη κατηγοριοποίηση, η οποία περιλαμβάνει:

1. **Isolated Sign Language Recognition (ISLR)** - Απομονωμένη Αναγνώριση Νοηματικής Γλώσσας
2. **Continuous Sign Language Recognition (CSLR**) - Συνεχής Αναγνώριση Νοηματικής Γλώσσας
3. **Sign Language Translation (SLT)** - Μετάφραση νοηματικής γλώσσας

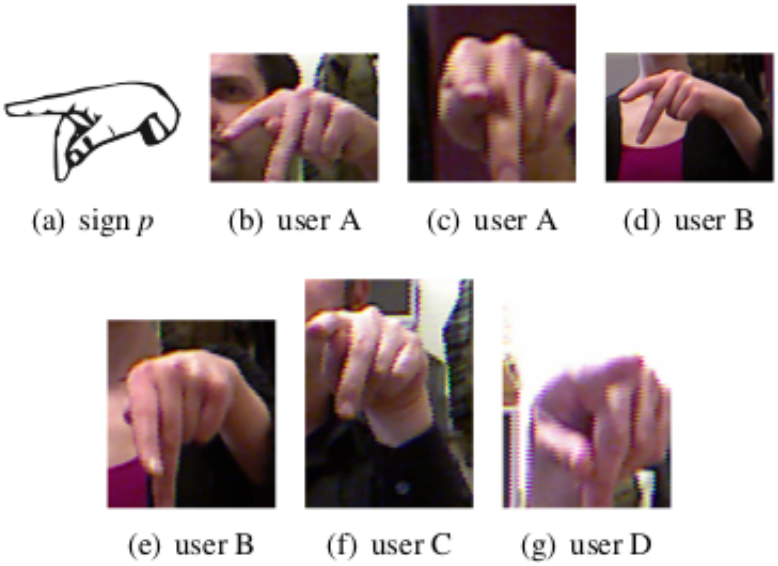
3.1 Απομονωμένη Αναγνώριση Νοηματικής Γλώσσας (ISLR)

Η απομονωμένη αναγνώριση νοηματικής γλώσσας έχει στόχο την αναγνώριση και μετάφραση μεμονωμένων νοημάτων και αποτελεί ουσιαστικά παρακλάδι του πεδίου της αναγνώρισης χειρονομιών (gesture recognition). Για αυτό το λόγο, είναι απαραίτητο ένα σύνολο δεδομένων το οποίο περιλαμβάνει αυστηρά χωρισμένα νοήματα και ετικέτες-στόχους σε αναλογία ένα-προς-ένα. Είναι η πρώτη μεθοδολογία που υλοποιήθηκε για τον σκοπό της μετάφρασης της νοηματικής και έχει συνεισφέρει πολλά στον τομέα, αλλά τα τελευταία χρόνια έχει παραγκωνιστεί λόγω των αυστηρών και χρονοβόρων απαιτήσεων που υπάρχουν, για να δημιουργηθεί ένα σύνολο δεδομένων κατάλληλο για καθημερινή χρήση.

Από τις πρώτες προσπάθειες, έγιναν το 1995 [5] όπου οι συγγραφείς χρησιμοποίησαν κάμερα για να εξάγουν τις τροχιές των χεριών και σε συνδυασμό με ένα μοντέλο Hidden Markov Model (HMM), να αναγνωρίσουν νοήματα από ένα σετ 40 λέξεων της Αμερικανικής Νοηματικής Γλώσσας (American Sign Language - ASL). Τέτοιου είδους χαρακτηριστικά όμως, είναι χρονοβόρο να φτιαχτούν γιατί πρέπει να οριστούν εκ των προτέρων, σε αντίθεση με τα συνελικτικά νευρωνικά δίκτυα, που εξάγουν χαρακτηριστικά αυτόματα. Επίσης είναι δύσκολο να αξιοποιηθούν σε πραγματικό περιβάλλον όπου η εικόνα στο βίντεο περιλαμβάνει πληθώρα χαρακτηριστικών, κάτι που φαίνεται και από το γεγονός ότι πέτυχαν καλύτερα αποτελέσματα όταν τα χέρια ήταν καλυμμένα με γάντια σκούρου χρώματος. Άλλες δουλειές όπως στο [6], αξιοποιούν αισθητήρες σε γάντια προκειμένου να κάνουν την κατηγοριοποίηση, πάλι όμως με παρόμοια προβλήματα. Αργότερα, το 2011, με τη δημιουργία του microsoft kinect, οι Pugeault και Bowden, αξιοποίησαν την επιπλέον διάσταση βάθους που προσέφερε για πιο αποτελεσματική αναγνώριση, μειώνοντας την επίδραση του υπόβαθρου [7]. Λίγο αργότερα, στο [8] χρησιμοποιήθηκε και πάλι το kinect και η διάσταση βάθους που προσφέρει αλλά αυτή τη φορά σε συνδυασμό με CNN για την εξαγωγή χαρακτηριστικών.

Στη συνέχεια όμως, και παρά ορισμένα μεγάλα σύνολο δεδομένων που δημοσιεύτηκαν για τον σκοπό του ISLR όπως το ASL Finger spelling Dataset [7] και το Word-level ASL dataset [9], η εξέλιξη της

τεχνολογίας και συγκεκριμένα η άνοδος της βαθιάς μάθησης και των προηγμένων καρτών γραφικών, έθεσε τα θεμέλια για την εξερεύνηση νέων, πιο αποδοτικών και υποσχόμενων μεθόδων.



Σχήμα 3.1: [7] Τα μοντέλα μαθαίνουν να αναγνωρίζουν κάποιο μεμονωμένο γράμμα ή λέξη αντιστοιχίζοντας χαρακτηριστικά από συγκεκριμένα frames σε αυτό

3.2 Συνεχής Αναγνώριση Νοηματικής Γλώσσας (CSLR)

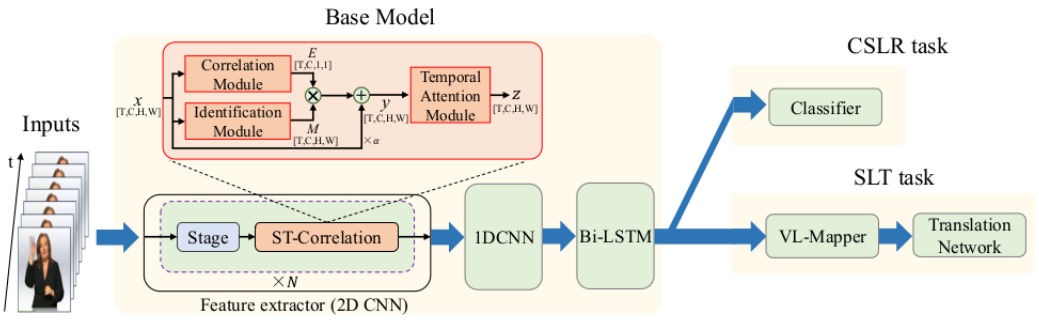
Σε αντίθεση με την ISLR, η συνεχής αναγνώριση νοηματικής γλώσσας αντιμετωπίζει τη μετάφραση της νοηματικής με πιο φυσικό τρόπο, χρησιμοποιώντας για την εκπαίδευση ολόκληρες φράσεις αντί για μεμονωμένες λέξεις. Αναλυτικότερα, τα σύνολα δεδομένων CSLR, περιλαμβάνουν μία σειρά από καρέ ως είσοδο, που συνθέτουν ένα βίντεο μίας φράσης στη νοηματική, ενώ η αντίστοιχη ετικέτα-στόχος είναι μία σειρά από γλωσσικές μονάδες (glosses). Οι γλωσσικές μονάδες χρησιμοποιούνται ως η πιο κοντινή απόδοση της σημασίας ενός νοήματος στον γραπτό λόγο. Στο μοντέλο δε παρέχονται χρονικές πληροφορίες για να συσχετίσει κάποια καρέ με κάποιες γλωσσικές μονάδες. Αντίθετα, πρέπει μόνο του να μάθει τις χρονικές συσχετίσεις και εξαρτίσεις μεταξύ λέξεων και καρέ, και πρέπει να τα μάθει παρά τις διαφορές στην ταχύτητα και τα στυλ της νοηματικής από διερμηνέα σε διερμηνέα.

Οι πρώτες εργασίες που χρησιμοποίησαν ολόκληρες προτάσεις για μετάφραση, έκαναν την εμφάνιση τους στη περίοδο 1990-1995 και έκαναν χρήση Hidden Markov Models (HMM) [10] και παράλληλων HMM [11] για την χρονική μοντελοποίηση των ακολουθιών, σε συνδυασμό με χαρακτηριστικά που είχαν ορίσει εκ των προτέρων. Η δουλειά πάνω στα HMM συνεχίστηκε [12], [13] αλλά η επιτυχία των CNN στην επεξεργασία εικόνας και των RNN στην επεξεργασία γλώσσας, δεν άργησε να περάσει και στην έρευνα για την μετάφραση της νοηματικής.

Η πιο συχνή αρχιτεκτονική, που χρησιμοποιείται έως και σήμερα, περιλαμβάνει έναν εξαγωγέα χαρακτηριστικών, με τον οποίο εξάγεται πληροφορία είτε για ένα μεμονωμένο είτε για γειτονικά καρέ, κι ένα κομμάτι υπεύθυνο για τη χρονική μοντελοποίηση. Ο εξαγωγέας χαρακτηριστικών μπορεί να είναι 2D

[14], 3D [15] ή 2+1D CNN [16]. Έπειτα ακολουθεί ένα LSTM και τέλος, για την εκμάθηση ακολουθιών μπορούν να χρησιμοποιηθούν κυρίως HMM [14] ή CTC [17]–[20], το οποίο χρησιμοποιείται εκτενώς στην βιβλιογραφία, για λόγους που θα αναλυθούν σε επόμενο κεφάλαιο. Κάτι που συναντάται επίσης συχνά, είναι η χρήση πολυτροπικών μέσων, όπως στο [21] που εξάγονται χαρακτηριστικά σημεία του ανθρώπινου σώματος κατά την είσοδο στο μοντέλο και χρησιμοποιούνται στην εκμάθηση, ώστε να είναι πιο εύκολη η παρακολούθηση σημείων βαρύτητας, όπως τα χέρια και το πρόσωπο.

Οι βασικοί τρόποι δημιουργίας νέων μοντέλων είναι είτε η βελτίωση του εξαγωγέα χαρακτηριστικών [19] είτε η εισαγωγή κάποιας επιπλέον συνάρτησης απώλειας στο κομμάτι της εκμάθησης ακολουθιών [16], χωρίς όμως να περιορίζονται σε αυτούς. Το μοντέλο με τα καλύτερα αποτελέσματα αυτή τη στιγμή (state-of-the-art) είναι το [17] με Word Error Rate (WER) 18.2% στο σύνολο τεστ, το οποίο ακολουθεί τη πρώτη τακτική. Συγκεκριμένα προσθέτει μία μονάδα αναγνώρισης, η οποία έχει σκοπό τη βελτίωση της αναγνώρισης του σώματος μέσα στο καρέ και μία μονάδα συσχέτισης, η οποία συσχετίζει γειτονικά καρέ για να εξάγει πληροφορία σχετική με τις τροχιές που ακολουθούν σημεία βαρύτητας, όπως τα χέρια. Τέλος, η πληροφορία αυτή διέρχεται από μία *χρονική μονάδα προσοχής* (temporal attention module), στην οποία καθορίζονται βάρη για τα καρέ ανάλογα με την ποσότητα πληροφορίας που φέρουν, δηλαδή, το πόσο σημαντικά είναι για την εκπαίδευση.



Σχήμα 3.2: [17] Η αρχιτεκτονική του state-of-the-art μοντέλου στην συνεχή αναγνώριση νοηματικής γλώσσας, CorrNet+

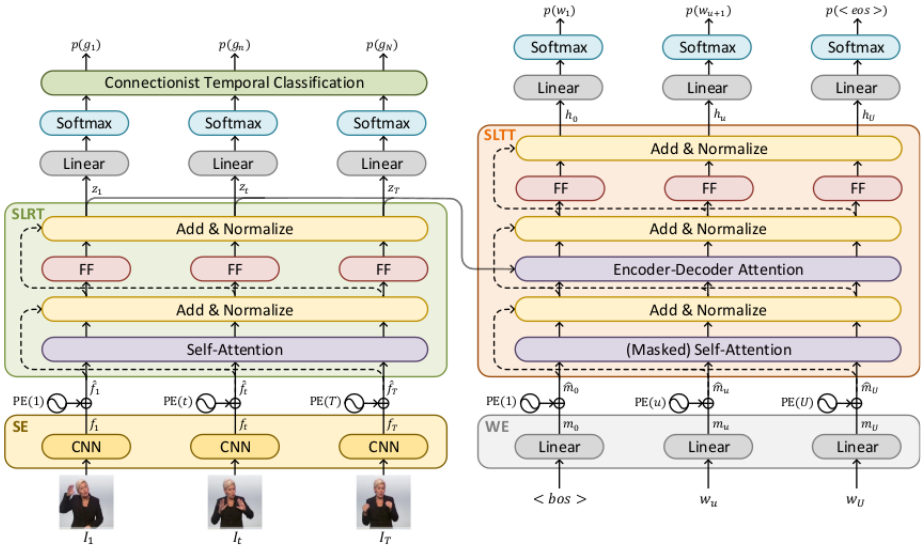
3.3 Μετάφραση Νοηματικής Γλώσσας (SLT)

Η μετάφραση νοηματικής αποτελεί μία κατηγορία μεθοδολογιών, αλλά παράλληλα είναι ο τελικός στόχος που προσπαθεί να επιτύχει αυτό το πεδίο έρευνας. Είναι, δηλαδή, η διαδικασία κατά την οποία ένα βίντεο μιας φράσης σε νοηματική, μπορεί να μετατραπεί ορθώς σε μια φράση σε γραπτή γλώσσα. Αυτή η μέθοδος χωρίζεται σε κάποιες υποκατηγορίες, οι οποίες εξαρτώνται από τον τρόπο και τα ενδιάμεσα στάδια, αν υπάρχουν, που το μοντέλο καταλήγει από το νόημα στην πλήρη φράση. Αυτές είναι [22]:

- **Sign2Text (S2T)**: Περιλαμβάνει τη διαδικασία από άκρη σε άκρη. Η μετάφραση γίνεται από τα καρέ του βίντεο, κατευθείαν σε γραπτή γλώσσα.
- **Sign2Gloss → Gloss2Text (S2G → 2T)**: Πρόκειται, ουσιαστικά, για τον συνδυασμό δύο μοντέλων. Το πρώτο είναι ένα μοντέλο όπως αυτά που αναφέρθηκαν παραπάνω, το οποίο λαμβάνει ως είσοδο καρέ και εξάγει μία σειρά από γλωσσικές μονάδες. Το δεύτερο είναι ένα μοντέλο το οποίο

έχει προεκπαιδευτεί με τις γλωσσικές μονάδες και τις αντίστοιχες τους ολοκληρωμένες φράσεις, παίρνει ως εισόδους την ακολουθία γλωσσικών αποδόσεων και εξάγει τις μεταφράσεις.

- **Sign2Gloss2Text (S2G2T)**: Όπως το προηγούμενο, αλλά το δεύτερο μοντέλο δεν είναι προεκπαιδευμένο, αλλά εκπαιδεύεται εκείνη τη στιγμή.
- **Sign2(Gloss+Text) (S2(G+T))**: Πάλι αποτελείται από τα ίδια δύο επιμέρους μοντέλα αλλά η εκπαίδευση γίνεται από άκρο σε άκρο, όπως στο S2T και οι γλωσσικές μονάδες χρησιμοποιούνται για επιπλέον επίβλεψη του μοντέλου. [21]



Σχήμα 3.3: [23] Παράδειγμα αρχιτεκτονικής S2T μοντέλου

Αξίζει να σημειωθεί ότι πολλά μοντέλα έχουν τη δυνατότητα να εκπαιδευτούν με διάφορους από τους προαναφερθέντες τρόπους [17], [23], [24]. Μάλιστα, η δοκιμή για το ποια από αυτές τις μεθόδους θα αποδώσει τα καλύτερα αποτελέσματα αποτελεί συχνή πρακτική.

3.4 Σύνολα δεδομένων

Στην εξέλιξη του κλάδου της αναγνώρισης και μετάφρασης νοηματικής γλώσσας συνέβαλαν σημαντικά, εκτός από τα μοντέλα συνελικτικών δικτύων και τα νέα σύνολα δεδομένων που δημοσιεύτηκαν, όπως αναφέρθηκε και παραπάνω. Στο κομμάτι που ακολουθεί θα περιγραφούν αναλυτικά αυτά που αξιοποιήθηκαν κατά την εκπόνηση αυτής της εργασίας.

- **RWTH-PHOENIX-Weather-2014 [25]**: Αποτελεί σύνολο δεδομένων ορόσημο στην έρευνα για την αναγνώριση νοηματικής γλώσσας και εμφανίζεται από το 2014 έως και σήμερα ως σημείο αναφοράς νέων μοντέλων. Τα βίντεο προέρχονται από ένα γερμανικό κανάλι καιρού, οπότε οι καταγεγραμμένες φράσεις σχετίζονται κυρίως με τον καιρό και τις μετεωρολογικές προβλέψεις στη γερμανική γλώσσα. Επίσης σημαίνει πως καταγράφηκαν σε πραγματικές συνθήκες, αποτυπώνοντας ρεαλιστικά εκφράσεις και τη ροή του λόγου του διερμηνέα. Τα βίντεο, λόγω της αναμετάδοσης, έχουν ανάλυση 210×260 εικονοστοιχεία και είναι καταγεγραμμένα με 25 καρέ ανά δευτερόλεπτο (frames per second). Σχετικά με τη μετάφραση, παρέχονται φράσεις που αποτελούνται από

γλωσσικές μονάδες, οι οποίες αντιστοιχίζονται σε ένα βίντεο-φράση. Για το σύνολο παρέχονται δύο διατάξεις, μία που περιλαμβάνει πολλαπλούς διερμηνείς και μία που περιλαμβάνει μόνο έναν. Η πρώτη, περιλαμβάνει συνολικά 9 διερμηνείς, με 12,5 περίπου ώρες συνολικού βίντεο, περίπου 6800 προτάσεις και σύνολο γύρω στις 1500 μοναδικές γλωσσικές μονάδες. Αυτά τα δεδομένα, τα χωρίζει στα τρία για το σκοπό της εκπαίδευσης, στο σύνολο εκπαίδευσης, το σύνολο αξιολόγησης και το σύνολο ελέγχου (train, dev, test sets). Η δεύτερη διάταξη που παρέχεται, περιλαμβάνει τα βίντεο με τις αποδόσεις μόνο ενός διερμηνέα. Αυτό μειώνει σημαντικά το μέγεθος του συνόλου και των μοναδικών γλωσσικών αποδόσεων και για αυτό η προηγούμενη διάταξη είναι αυτή που χρησιμοποιείται πιο συχνά.

	PHOENIX SingleSigner		**PHOENIX MultiSigner**		
	Train	Test	Train	Dev	Test
# διερμηνείς	1	1	9	9	9
διάρκεια [ώρες]	0,51	0,0075	10,71	0,84	0,99
# καρέ	46.282	6.751	963.664	75.186	89.472
# προτάσεις	304	47	5,672	540	629
μέγεθος λεξιλογίου	266	-	1.081	467	500
εκτός λεξιλογίου [%]	-	1,6	-	0,5	0,54

Πίνακας 3.1: Βασικά χαρακτηριστικά του συνόλου δεδομένων RWTH-PHOENIX-Weather.

- **Greek Sign Language Dataset (GSL)[26]**: Το GSL δημοσιεύτηκε το 2020 και αποτελεί το πρώτο σύνολο δεδομένων που αφορά την συνεχή αναγνώριση νοηματικής γλώσσας για την ελληνική γλώσσα. Αντίθετα με το προηγούμενο σύνολο δεδομένων, οι φράσεις είναι προδιαγεγραμμένες και κάθε φράση την εκτελεί καθένας από τους διερμηνείς. Οι φράσεις περιλαμβάνουν σενάρια κατά τα οποία ένα άτομο αλληλεπιδρά με άλλα στα πλαίσια δημόσιων υπηρεσιών (αστυνομία, νοσοκομείο, κέντρα εξυπηρέτησης πολιτών). Η καταγραφή έγινε σε 30 frames per second, με ανάλυση 848×480 και χρησιμοποιήθηκε κάμερα RGB+D. Έτσι, εκτός από τις εικόνες, παρέχονται και δεδομένα βάθους.

Σχήμα 3.4: [26] Παράδειγμα καρέ από το GSL

Συνολικά, περιλαμβάνονται 10.295 προτάσεις με 310 μοναδικές γλωσσικές μονάδες από 7 διερμηνείς. Όπως και πριν, υπάρχουν πάλι διαφορετικές διατάξεις. Η πρώτη ονομάζεται ”Signer Dependent - SD” και χρησιμοποιεί έναν διαχωρισμό σε σύνολα train-dev-test με ποσοστά 80%, 10% και 10% του συνόλου των προτάσεων αντίστοιχα. Σε αυτό τον διαχωρισμό, το σύνολο test έχει φράσεις που δεν υπάρχουν στα υπόλοιπα, αλλά οι γλωσσικές μονάδες που χρησιμοποιεί είναι

όλες γνωστές. Η επόμενη διάταξη είναι η ”Signer Independent - SI” στην οποία το μοντέλο εκπαιδεύεται σε όλα τα βίντεο από τους 6 διερμηνείς, ενώ τα βίντεο αυτού που απομένει χωρίζονται στα σύνολα dev και test. Το GSL επίσης, εκτός από μετάφραση σε γλωσσικές μονάδες, παρέχει και μετάφραση στην ελληνική γλώσσα, καθιστώντας το χρήσιμο και για τη κατηγορία μετάφρασης νοηματικής γλώσσας. Στο σχήμα 3.5 φαίνεται πως είναι δομημένες οι μεταφράσεις. Τέλος, παρέχεται ακόμα μία διάταξη η οποία είναι κατάλληλη για απομονωμένη αναγνώριση νοηματικής γλώσσας, την οποία την ονομάζουν ”GSL isolated”.

kep4_signer1_rep3_sentences/sentences0007|γράψτε τη σημερινή ημερομηνία|ΕΣΥ ΗΜΕΡΑ ΜΗΝΑΣ ΣΗΜΕΡΑ ΓΡΑΦΩ

Σχήμα 3.5: Παράδειγμα μετάφρασης σε γλωσσικές μονάδες και στην ελληνική γλώσσα από το GSL. Το πρώτο κομμάτι υποδηλώνει τον φάκελο που βρίσκονται τα καρέ του βίντεο με τη φράση, το δεύτερο είναι η μετάφραση στην ελληνική γλώσσα και το τελευταίο είναι η μετάφραση σε γλωσσικές μονάδες.

	GSL SI	GSL SD
# διερμηνείς	7	7
# προτάσεις	10.295	10.295
μέγεθος λεξιλογίου	310	310
διάρκεια	9,59	9,59
ανάλυση	848 × 480	848 × 480
fps	30	30

Πίνακας 3.2: Βασικά χαρακτηριστικά του συνόλου δεδομένων Greek Sign Language.

- **ImageNet [27]** : Το σύνολο δεδομένων Imagenet δημοσιεύτηκε το 2009 και αποτελείται από 14.197.122 εικόνες αντικειμένων με τις αντίστοιχες ετικέτες. Περιέχει συνολικά πάνω από 20.000 κατηγορίες και έχει αποτελέσει σύνολο ορόσημο στον τομέα της υπολογιστικής όρασης. Από το 2010 και μετά καθιέρωσε τον ετήσιο διαγωνισμό ILSVRC, στον οποίο εμφανίστηκαν και αναδείχτηκαν πολλά καινοτόμα μοντέλα που δοκιμάστηκαν στον εντοπισμό αντικειμένων. Επιπλέον, χρησιμοποιείται πολύ συχνά για την προεκπαίδευση μοντέλων, ώστε να αποκτήσουν μια ”εμπειρία” πάνω στην γενική αναγνώριση αντικειμένων και να μάθουν πιο εύκολα κάποια πιο εξειδικευμένη εργασία, όπως στη προκειμένη περίπτωση, η συνεχής αναγνώριση νοηματικής γλώσσας.

Κεφάλαιο 4ο: Θεωρητικό Υπόβαθρο

4.1 Μηχανική Μάθηση

4.1.1 Εισαγωγή

Μάθηση είναι διαδικασία κατά την οποία ένα σύστημα βελτιώνει την επίδοση του σε σχέση με μία διεργασία, καθώς αυξάνεται η εμπειρία του σχετικά με αυτή την διεργασία [28]. Αυτό σημαίνει πως ένα σύστημα βασιζόμενο στα δεδομένα που έχει, κάνει κάποια πρόβλεψη και ανάλογα με το πόσο σωστή ή λάθος είναι αυτή, προσαρμόζεται με σκοπό να τη βελτιώσει. Κάποια χαρακτηριστικά παραδείγματα μάθησης αποτελούν:

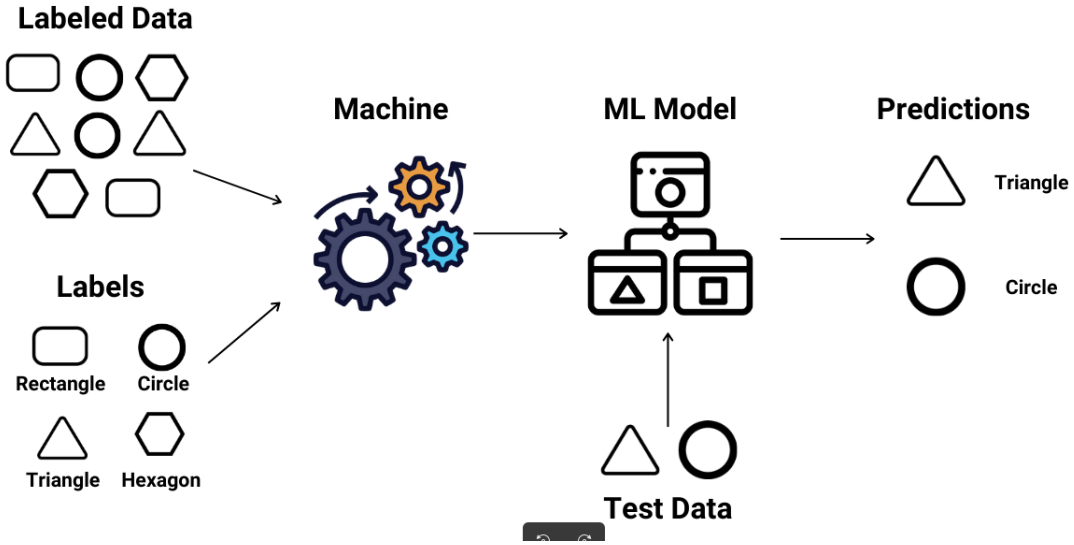
- Ταξινόμηση: πχ. Η ορθή αναγνώριση ενός αντικειμένου σε περιβάλλον με διάφορα άλλα
- Παλινδρόμηση (πρόβλεψη συνεχών τιμών): πχ. Η πρόβλεψη αξίας κατοικιών, βασισμένο σε δεδομένα όπως η αξία των κατοικιών στη γύρω περιοχή και χαρακτηριστικά της ίδιας της κατοικίας όπως μέγεθος, αριθμός μπάνιων κ.ά.
- Ομαδοποίηση: πχ. Η κατηγοριοποίηση πελατών με βάση τις καταναλωτικές τους συνήθειες
- Εντοπισμός ανωμαλιών: πχ. Αναγνώριση καρκινικού κυττάρου ανάμεσα σε κανονικά
- Συμπίεση δεδομένων: πχ. Η απλοποίηση ενός συνόλου δεδομένων στα πιο βασικά του χαρακτηριστικά με σκοπό την μείωση παραμέτρων με τη λιγότερη δυνατή απώλεια πληροφορίας
- Ανάπτυξη στρατηγικών: πχ. Η ανάλυση κινήσεων και ολόκληρων παιχνιδιών σκακιού ώστε να βρεθεί η βέλτιστη κίνηση κάθε δεδομένη στιγμή.

Η μηχανική μάθηση, λοιπόν έχει ως στόχο τη δημιουργία αλγορίθμων οι οποίοι είναι ικανοί να αυτοβελτιώνονται βασισμένοι στα δεδομένα που έχουν ως είσοδο με στόχο την επίλυση προβλημάτων όπως αυτά που αναφέρθηκαν παραπάνω. Μία πολύ σημαντική διευκρίνηση που αξίζει να γίνει, είναι πως ο στόχος της μηχανικής μάθησης δεν είναι η απομνημόνευση των δεδομένων και δυνατότητα αναπαραγωγής σωστών απαντήσεων μόνο πάνω σε αυτά τα δεδομένα. Αντίθετα, σκοπός είναι η σωστή αξιοποίηση των δεδομένων εισόδου ώστε να γίνει ορθή εκτίμηση για δεδομένα που ο αλγόριθμος δεν έχει αντιμετωπίσει στο παρελθόν. Για παράδειγμα, εάν ο σκοπός ενός αλγορίθμου είναι η αναγνώριση της γάτας ανάμεσα σε άλλα ζώα και αντικείμενα, δεν είναι επιθυμητό ο αλγόριθμος αυτός να μπορεί να εντοπίζει μόνο γάτες ίδιου χρώματος ή μεγέθους με αυτές που έχει δει κατά τη διάρκεια της εκπαίδευσης του, αλλά να μπορεί να εντοπίζει όλες τις γάτες.

4.1.2 Είδη μάθησης

Η μηχανική μάθηση χωρίζεται σε τρία βασικά είδη, τη μάθηση με επίβλεψη, τη μάθηση χωρίς επίβλεψη και τη μάθηση με ενίσχυση.

4.1.2.1 Μάθηση με επίβλεψη Στη μάθηση με επίβλεψη (supervised learning), τα δεδομένα εισόδου του μοντέλου αποτελούνται από τα πρότυπα $(x_1, x_2, x_3, \dots)$ και τους στόχους/ετικέτες $(t_1, t_2, t_3, \dots)$. Με βάση αυτές τις εισόδους, ο αλγόριθμος προσπαθεί να αναγνωρίσει μοτίβα ή να εξάγει χαρακτηριστικά που διαφοροποιούν τις καταστάσεις ώστε να αυξήσει τα ποσοστά επιτυχίας του στην σύνδεση ενός προτύπου με την κατάλληλη ετικέτα. Αναλυτικότερα, ο αλγόριθμος αφού ολοκληρώσει την εκπαίδευση, είναι σε θέση να δέχεται κάποια πρότυπα εισόδου $(x'_1, x'_2, x'_3, \dots)$ και να παράγει ως έξοδο μία σειρά προβλέψεων $(y_1, y_2, y_3, \dots)$. Για να είναι επιτυχημένος ο αλγόριθμος, πρέπει οι ετικέτες των εισόδων να είναι όμοιες με τις προβλέψεις του $(t'_1, t'_2, t'_3, \dots)$. Παραδείγματα μάθησης με επίβλεψης αποτελούν τα προβλήματα ταξινόμησης και παλινδρόμησης που αναφέρθηκαν παραπάνω.



Σχήμα 4.1: Μάθηση με επίβλεψη

4.1.2.2 Μάθηση χωρίς επίβλεψη Αντίθετα με τον προηγούμενο τύπο μάθησης, στη μάθηση χωρίς επίβλεψη (unsupervised learning) η είσοδος στο μοντέλο αποτελείται μόνο από τα πρότυπα $(x_1, x_2, x_3, \dots)$, χωρίς στόχους/ετικέτες. Σκοπός του αλγορίθμου είναι και πάλι να αναγνωρίσει τα πιο σημαντικά χαρακτηριστικά των δεδομένων εισόδου, με σκοπό αυτή τη φορά να τα ομαδοποιήσει με τον καλύτερο και πιο αντιπροσωπευτικό τρόπο. Έτσι, όταν έρθουν καινούρια δεδομένα ο αλγόριθμος προσπαθεί να τα ταιριάξει σε κάποια από τις υπάρχουσες κατηγορίες ή να δημιουργήσει καινούρια. Σε αυτή τη κατηγορία αλγορίθμων ανήκουν τα παραδείγματα εντοπισμού ανωμαλιών, ομαδοποίησης και της συμπίεσης δεδομένων. Κάποιοι από τους πιο γνωστούς αλγορίθμους είναι οι K-means clustering, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) και PCA (Principle Component Analysis).

4.1.2.3 Μάθηση με ενίσχυση Στη μάθηση με ενίσχυση (reinforced learning) το μοντέλο δέχεται πάλι μόνο πρότυπα $(x_1, x_2, x_3, \dots)$ στην είσοδο του. Η διαφορά είναι πως υπάρχει ένα σύστημα επιβράβευσης/τιμωρίας για την επίτευξη του σκοπού της μάθησης. Ο αλγόριθμος, δηλαδή, κάνει διάφορες προβλέψεις ή κάνει κάποιες δράσεις και ανάλογα με το αν είναι επιθυμητή η πράξη που έκανε, έχει είτε θετική ανταμοιβή ή κάποια ποινή, τα οποία συνήθως εκφράζονται σε ένα σύστημα με πόντους. Ένα παράδειγμα τέτοιο αλγορίθμου μπορεί να είναι ένας αλγόριθμος που προσπαθεί να μάθει την βέλτιστη διαδρομή που

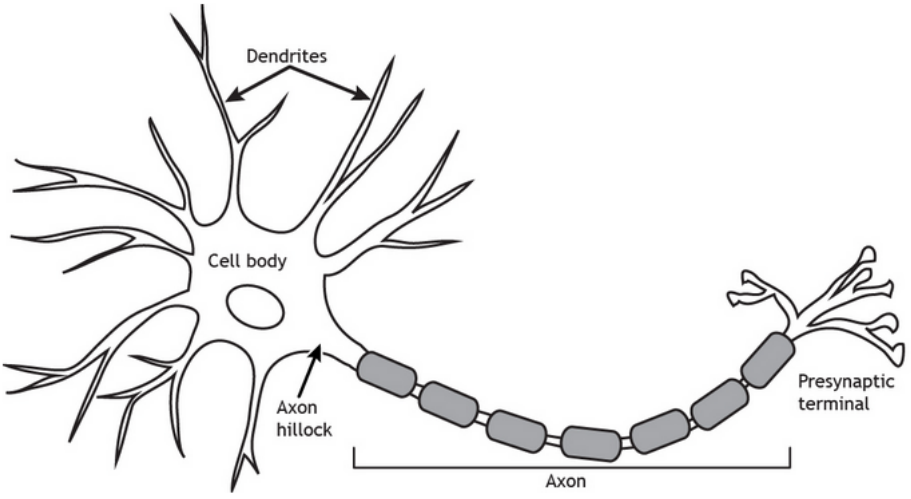
πρέπει να ακολουθήσει ένα αμάξι σε μία πίστα. Ο αλγόριθμος ξεκινάει να δοκιμάζει κάθε δυνατή διαδρομή στέλνοντας το αμάξι προς κάθε κατεύθυνση στην αρχή. Κάθε φορά που το αμάξι πηγαίνει προς τη θεμιτή κατεύθυνση θα επιβραβεύεται, ενώ αντίθετα θα αφαιρούνται πόντοι. Έτσι, μετά από πολλές επαναλήψεις θα βρεθεί η διαδρομή που πρέπει να ακολουθήσει το αμάξι. Η ανάπτυξη στρατηγικών που αναφέρθηκε παραπάνω ανήκει επίσης σε αυτή τη κατηγορία μάθησης.

4.2 Νευρωνικά Δίκτυα

Τα νευρωνικά δίκτυα, όπως φαίνεται κι από το όνομα τους, ξεκίνησαν εμπνευσμένα από τον τρόπο τρόπο που μαθαίνει και επεξεργάζεται τη πληροφορία ο εγκέφαλος. Συγκεκριμένα, έχουν εμπνευστεί από τους νευρώνες τους εγκεφάλου και τον τρόπο με τον οποίο λειτουργούν. Η έρευνα πάνω σε αυτόν τον τομέα ξεκίνησε από την προσπάθεια αναπαραγωγής με ψηφιακά μέσα, της λειτουργίας του εγκεφάλου και τη δημιουργία τεχνητής νοημοσύνης μέσω αλγορίθμων. Κατά συνέπεια, το πρώτο βήμα για την κατανόηση των νευρωνικών δικτύων είναι η εκμάθηση του τρόπου λειτουργίας ενός βασικού νευρώνα.

4.2.1 Νευρώνας

Οι νευρώνες αποτελούν το στοιχείο που διαδραματίζει τον μεγαλύτερο ρόλο στη διαδικασία της σκέψης, της μάθησης και και της επεξεργασίας πληροφορίας. Χωρίζονται σε διάφορα είδη ανάλογα με την λειτουργία που επιτελούν ή τη μορφολογία τους. Η γενική μορφολογία των νευρώνων περιλαμβάνει τους δενδρίτες, το κυρίως σώμα, τον άξονα και τις νευροαξονικές απολήξεις. Κύρια είδη νευρώνων είναι οι αισθητήριοι, οι ενδιάμεσοι και οι κινητήριοι οι οποίοι συμβάλλουν με διαφορετικές λειτουργίες ο καθένας, από την λήψη ερεθισμάτων, στη μεταφορά τους μέχρι τα δραστικά κύτταρα. Ουσιαστικά, η κύρια λειτουργία των νευρώνων είναι η μεταφορά ηλεκτρικών σημάτων (παλμών) σε μικρές και μεγάλες αποστάσεις. Αυτό επιτυγχάνεται με την αλληλεπίδραση μεταξύ των νευρώνων, η οποία οδηγεί στη δημιουργία ενός νευρωνικού δικτύου. Οι δενδρίτες λαμβάνουν την πληροφορία από άλλους νευρώνες, η οποία επεξεργάζεται στο κυρίως σώμα και μεταφέρεται μέσω του άξονα στις συνάψεις απ' όπου μεταφέρεται στον επόμενο νευρώνα.



Σχήμα 4.2: [29] Ο νευρώνας

4.2.2 Πρώτος ψηφιακός νευρώνας

Η πρώτη απόπειρα μεταφοράς του βιολογικού νευρώνα στον ψηφιακό τομέα, πραγματοποιήθηκε με επιτυχία από τους McCulloch και Pitts το 1943 [30] οι οποίοι περιόρισαν την λειτουργία του νευρώνα στα πολύ βασικά του κομμάτια. Συγκεκριμένα, όρισαν τη κατάσταση του νευρώνα ως ένα δυαδικό σύστημα σύμφωνα με το οποίο ο νευρώνας είναι είτε ενεργός και πυροδοτεί με μέγιστη συχνότητα παλμών, είτε είναι ανενεργός. Για την αλλαγή της κατάστασης του νευρώνα ακολουθείται η παρακάτω λογική:

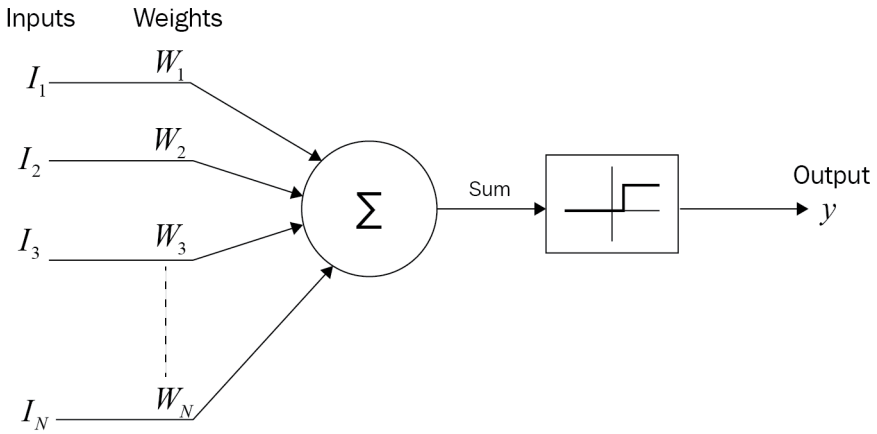
- Για εισόδους $(x_1, x_2, x_3, \ldots, x_n)$ και συναπτικά βάρη $(w_1, w_2, w_3, \ldots, w_n)$, ελέγχουμε εάν το επιμέρους άθροισμα των γινομένων των εισόδων με τα βάρη είναι μεγαλύτερο από μία τιμή κατωφλίου θ .
- Αν αυτό ισχύει, ο νευρώνας ενεργοποιείται, αλλιώς παραμένει αδρανής

Οπότε, η μαθηματική έκφραση της διέγερσης του νευρώνα είναι

$$u = \sum_{i=1}^{n} (w_i x_i - \theta) \tag{4.2.1}$$

$$y = f(u) = \begin{cases} 0 & \text{αν } u \leq 0, \\ 1 & \text{αν } u > 0 \end{cases} \tag{4.2.2}$$

Όπου y είναι η κατάσταση του νευρώνα, u η διέγερση και $f(\cdot)$ η βηματική συνάρτηση στη προκειμένη περίπτωση, αλλά γενικά μπορούν να χρησιμοποιηθούν διάφορες συναρτήσεις ενεργοποίησης, τις πιο γνωστές από τις οποίες θα δούμε παρακάτω. Να σημειωθεί πως το κατώφλι $theta$ και τα βάρη w_i είναι πραγματικοί αριθμοί και τα βάρη αντιπροσωπεύουν στην ουσία έναν δείκτη ο οποίος δείχνει πόσο σημαντική είναι μία συγκεκριμένη είσοδος.



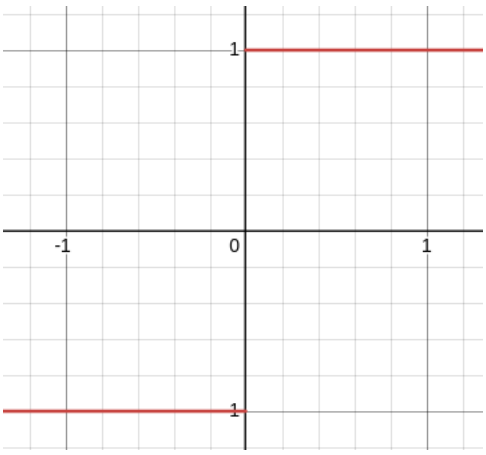
Σχήμα 4.3: [31] Μοντέλο McCullogh-Pits

4.2.2.1 Συναρτήσεις Ενεργοποίησης Εκτός από το βασικό μοντέλο των McCullogh-Pits έχουν δημιουργηθεί διάφορα ακόμα μοντέλα. Μια σημαντική διαφοροποίηση αυτών είναι πως μπορούν να χρη-

σιμοποιούν διαφορετική συνάρτηση προκειμένου να μοντελοποιήσουν την συνάρτηση ενεργοποίησης του νευρώνα. Κάποιες από τις πιο διαδεδομένες είναι:

- Η βηματική συνάρτηση (-1/1)

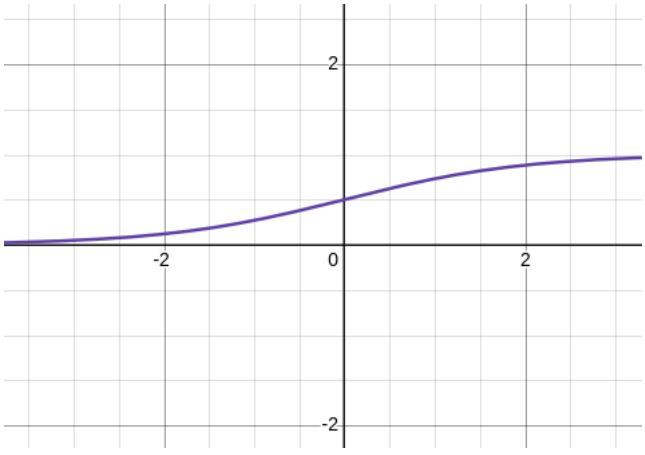
$$f(u) = \begin{cases} -1 & \text{αν } u \leq 0, \\ 1 & \text{αν } u > 0 \end{cases} \tag{4.2.3}$$



Σχήμα 4.4: Βηματική Συνάρτηση (Step Function)

- Η σιγμοειδής συνάρτηση

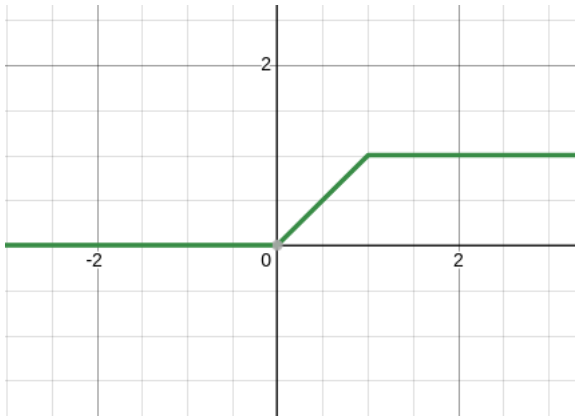
$$f(u) = \frac{1}{1 + e^{-u}} \tag{4.2.4}$$



Σχήμα 4.5: Σιγμοειδής Συνάρτηση (Sigmoid Function)

- Η συνάρτηση κατωφλίου

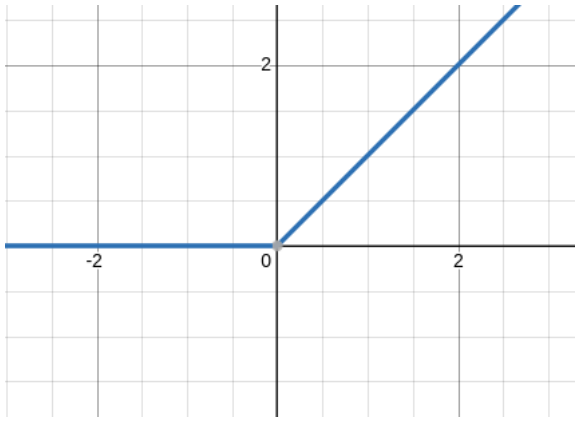
$$f(u) = \begin{cases} 0 & \text{αν } u \leq 0, \\ u & \text{αν } 0 < u < 1, \\ 1 & \text{αν } u \geq 1 \end{cases} \tag{4.2.5}$$



Σχήμα 4.6: Συνάρτηση Κατωφλίου (Threshold Function)

- Η συνάρτηση ράμπας (ReLU)

$$f(u) = \begin{cases} -1 & \text{αν } u \leq 0, \\ u & \text{αν } u > 0 \end{cases} \tag{4.2.6}$$



Σχήμα 4.7: Συνάρτηση Ράμπας (Rectified Linear Unit - ReLU)

4.2.3 Perceptron

Το δίκτυο Perceptron [32] αποτελείται από έναν μόνο νευρώνα McCullogh-Pits ο οποίος παράγει μία έξοδο βασισμένη στην είσοδο του, στα βάρη του και στον αριθμό κατωφλίου (ή πόλωση), σύμφωνα με την συνάρτηση:

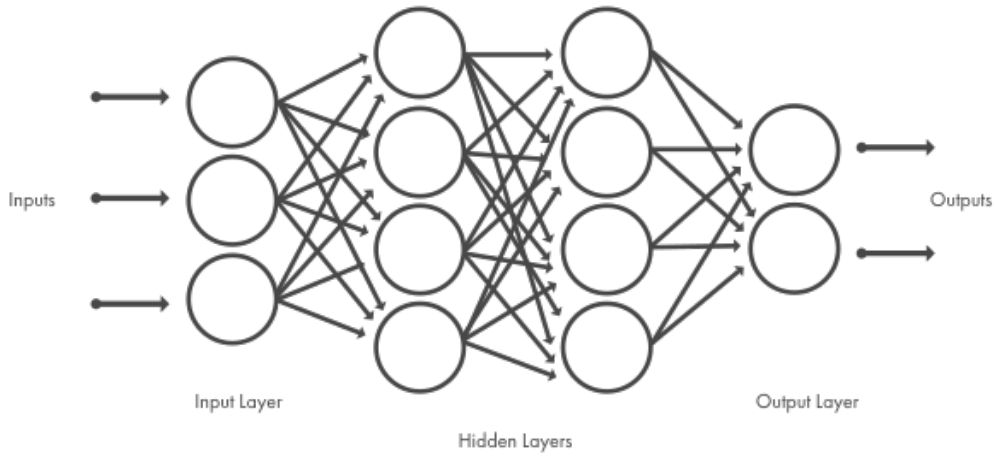
$$y = f(\sum_{i=1}^{n}(w_i x_i + b)) \tag{4.2.7}$$

Αυτή, βασίζεται στην εξ. 4.2.1 από νωρίτερα. Στόχος του δικτύου είναι να βελτιστοποιήσει τα βάρη w_i ώστε οι έξοδοι y_i να είναι όσο πιο κοντά γίνεται στους στόχους t_i. Για να το επιτύχει αυτό, κατά τη διάρκεια της εκπαίδευσης ελέγχει τη διαφορά της υπόθεσης του σε σχέση με το στόχο και ανανεώνει τα βάρη του κατάλληλα. Κάθε φορά που ο αλγόριθμος επεξεργάζεται όλα τα δεδομένα εισόδου που του τροφοδοτούνται και ανανεώνει τα βάρη του, ολοκληρώνει μία *εποχή* εκπαίδευσης. Ένα μοντέλο μπορεί να χρειάζεται να εκπαιδευτεί για αρκετές εποχές μέχρι να επιτύχει το επιθυμητό αποτέλεσμα. Από μαθηματικής άποψης, αυτό που προσπαθεί να κάνει ο αλγόριθμος είναι, χρησιμοποιώντας δεδομένα,

να ανακαλύψει μία γραμμική συνάρτηση τύπου $y = ax + b$ η οποία τα μοντελοποιεί. Επειδή, όμως λειτουργεί έτσι, ο αλγόριθμος έχει επιτυχία μόνο σε γραμμικά μοντέλα, όπου το πρόβλημα είναι γραμμικά διαχωρίσιμο. Σε περιπτώσεις που το πρόβλημα είναι μη γραμμικό, τότε το μοντέλο αποτυγχάνει και χρειάζεται κάποια διαφορετική λύση.

4.3 Βαθιά Μάθηση

Αντίθετα με το μοντέλο Perceptron, που βασίζεται σε έναν μόνο νευρώνα, η βαθιά μάθηση στηρίζεται σε πλήθος νευρώνων και μάλιστα σε πλήθος στρωμάτων νευρώνων. Αυτοί οι νευρώνες αλληλεπιδρούν μεταξύ τους προκειμένου να δημιουργηθεί ένας πιο σύνθετος τρόπος επεξεργασίας της αρχικής πληροφορίας (των εισόδων), με σκοπό να αντιμετωπιστούν και μη γραμμικά προβλήματα. Τέτοια μοντέλα αποτελούνται από το στρώμα εισόδου, το στρώμα εξόδου και στο ενδιάμεσο n κρυφά στρώματα. Σε αυτή τη περίπτωση, το n είναι και το βάθος του μοντέλου. Τα βαθιά μοντέλα έχουν παρόμοιες δυνατότητες με ρηχά μοντέλα πολλών νευρώνων. Η διαφορά που διαχωρίζει τα ρηχά με τα βαθιά, είναι πως ένα βαθύ δίκτυο μπορεί να επιτύχει τα αποτελέσματα ενός ρηχού με λιγότερους νευρώνες, το οποίο μειώνει σημαντικά την υπολογιστική πολυπλοκότητα. Η βαθιά μάθηση γνώρισε και συνεχίζει να γνωρίζει άνθηση στη σημερινή εποχή λόγω της εμφάνισης νέων μοντέλων όπως τα συνελικτικά νευρωνικά δίκτυα (CNNs), την ελεύθερη διάδοση ποιοτικών συνόλων δεδομένων, αλλά και τεχνολογικών εξελίξεων που περιλαμβάνουν τη βελτίωση των καρτών γραφικών. Οι τελευταίες διαδραμάτισαν σημαντικό ρόλο καθώς επιτρέπουν την πραγματοποίηση ταχύτατων παράλληλων υπολογισμών, που είναι ιδανικό για πράξεις μεταξύ πινάκων και διανυσμάτων που γίνονται σε μία εκπαίδευση δικτύου.



Σχήμα 4.8: Ένα νευρωνικό δίκτυο δύο στρωμάτων

4.3.1 Συνελικτικά Νευρωνικά Δίκτυα

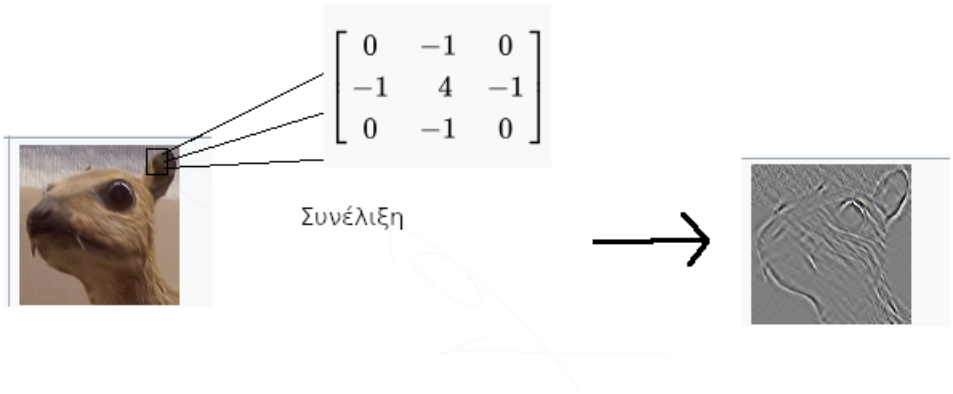
Τα συνελικτικά νευρωνικά δίκτυα (convolutional neural networks - CNNs), όπως αναφέρθηκε παραπάνω, είναι μοντέλα βαθιάς μάθησης, τα οποία χρησιμοποιούνται εκτενώς στον τομέα υπολογιστικής όρασης, ενώ έμπνευση για τη δομή τους αποτέλεσε και πάλι η βιολογία. Συγκεκριμένα, οι συγγραφείς του [33] πρότειναν για πρώτη φορά τη δομή των CNN βασισμένοι στη δουλειά των Hubel και Wiesel [34] πάνω στον οπτικό φλοιό της γάτας. Η αρχιτεκτονική των συνελικτικών δικτύων περιλαμβάνει την είσοδο, και στη συνέχεια μία αλληλουχία από στρώματα συνέλιξης (convolutional layers) και υποδειγματοληψίας

(pooling layers) τα οποία διαδέχονται το ένα μετά το άλλο. Τέλος, υπάρχει το στρώμα εξόδου ή στρώμα ταξινόμησης το οποίο είναι πλήρως συνδεδεμένο με το τελευταίο στρώμα και δρα ως ταξινομητής. Παρακάτω θα δούμε πως λειτουργούν αυτά τα στρώματα πιο αναλυτικά.

4.3.1.1 Συνελικτικό Στρώμα Το συνελικτικό στρώμα αποτελεί το κομμάτι των δικτύων όπου συμβαίνει η πιο θεμελιώδης διεργασία. Συγκεκριμένα, όταν στην είσοδο του μοντέλου τροφοδοτείται μια εικόνα, η είσοδος είναι στην ουσία ένας πίνακας τριών διαστάσεων $W \times H \times C$, όπου W είναι το πλάτος της εικόνας, H είναι το ύψος και C είναι το κανάλι. Το τελευταίο, για τη περίπτωση που η εικόνα είναι RGB, είναι ίσο με 3. Στη συνέχεια, δημιουργούνται *χάρτες χαρακτηριστικών* χρησιμοποιώντας τα *φίλτρα*. Τα φίλτρα είναι κι αυτά πίνακες τριών, συνήθως, διαστάσεων τα οποία περιέχουν κατάλληλα βάρη ώστε μέσω την συνέλιξης τους με την είσοδο, να εξαχθούν κάποια χαρακτηριστικά, τα οποία συνθέτουν τους χάρτες. Ένα παράδειγμα φαίνεται στο σχήμα 4.9 όπου ένα δισδιάστατο φίλτρο εφαρμόζεται σε μια εικόνα και το αποτέλεσμα είναι να απομονωθούν οι άκρες τις αρχικής εικόνας. Ο μαθηματικός τύπος σύμφωνα με τον οποίο παράγονται οι νευρώνες των χαρτών χαρακτηριστικών είναι ο εξής:

$$z_{i,j}^{(k)} = \sum_{c=0}^{C-1} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{i+m,j+n,c} \cdot w_{m,n,c}^{(k)} + b^{(k)} \tag{4.3.1}$$

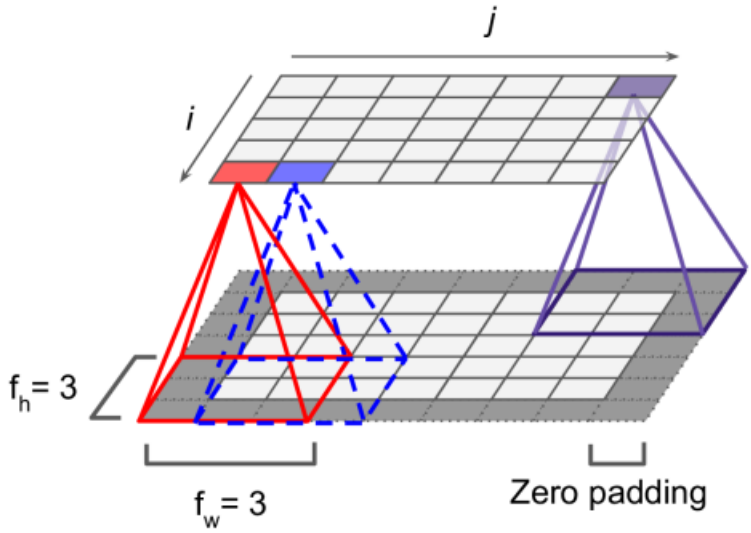
Όπου $z_{i,j}^{(k)}$ είναι ο νευρώνας που βρίσκεται στον k-οστό χάρτη χαρακτηριστικών στη θέση (i, j), $x_{i+m,j+n,c}$ είναι οι είσοδοι από την περιοχή $M \times N$ και $w_{m,n,c}^{(k)}$ είναι το φίλτρο ή μάσκα. Ο στόχος τους, μπορεί να συνοψιστεί ως η επιθυμία να προσαρμοστεί η εικόνα, ώστε να αναδειχτούν τα χαρακτηριστικά ενδιαφέροντος. Αυτό επιτυγχάνεται μέσω της κατάλληλης προσαρμογής των βαρών των φίλτρων.



Σχήμα 4.9: Το αποτέλεσμα της συνέλιξης εικόνας με φίλτρο δύο διαστάσεων για την αναγνώριση άκρων

Στη διαδικασία της συνέλιξης υπάρχουν τρεις βασικές παράμετροι οι οποίες επηρεάζουν τους χάρτες χαρακτηριστικών. Αυτές είναι:

- **Τα βάρη των φίλτρων:** Τα βάρη των φίλτρων προσαρμόζονται κατά τη διάρκεια της εκπαίδευσης, με στόχο να βρεθούν τα καταλληλότερα βάρη που εξάγουν τα επιθυμητά χαρακτηριστικά ανάλογα με το πρόβλημα που ζητείται να επιλυθεί. Ο τρόπος με τον οποίον προσαρμόζονται, είναι μέσω κάποιας συνάρτησης απώλειας (loss function), την οποία προσπαθεί να ελαχιστοποιήσει με τη χρήση κάποιου αλγορίθμου βελτιστοποίησης (πχ. stochastic gradient descent, adam) και τη χρήση οπισθοδιάδοσης (backpropagation). Επίσης, όπως φαίνεται από την εξίσωση 4.3.1, τα βάρη



Σχήμα 4.10: [35] Συνέλιξη με φίλτρο 3×3, μηδενικό γέμισμα και βήμα 1. Ο κάτω πίνακας είναι η είσοδος και ο πίνακας που βρίσκεται επάνω είναι ο χάρτης χαρακτηριστικών που δημιουργείται. Το κομμάτι της εισόδου που χρησιμοποιείται κάθε φορά ονομάζεται υποδεκτικό δίκτυο και είναι υπεύθυνο σε συνδυασμό με το φίλτρο για την τιμή που θα πάρει ο αντίστοιχος νευρώνας του χάρτη.

παραμένουν ίδια σε όλη τη δημιουργία των χαρτών χαρακτηριστικών, δηλαδή δεν επηρεάζονται από την θέση του νευρώνα (i, j). Αυτό έχει πολύ μεγάλο υπολογιστικό αντίκτυπο καθώς μειώνει σημαντικά τους υπολογισμούς που πρέπει να πραγματοποιηθούν για τις ανανεώσεις των βαρών.

- **Βήμα συνέλιξης (stride):** Στη περίπτωση που το βήμα της συνέλιξης είναι μεγαλύτερο του 1, το φίλτρο δεν εφαρμόζεται με τη σειρά σε όλη την είσοδο, αλλά κάθε φορά προσπερνάει θέσεις ίσες με το βήμα s. Η συνέλιξη με βήμα γίνεται σύμφωνα με τη σχέση

$$\text{Output}(i, j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} W_{m,n} \cdot X_{i \cdot s_i + m, j \cdot s_j + n} \tag{4.3.2}$$

  Η χρήση βήματος γίνεται προκειμένου να μειωθούν οι πράξεις, καθώς το φίλτρο χρειάζεται να εφαρμοστεί λιγότερες φορές, και συχνά έχει και μικρό αντίκτυπο στην επίδοση.

- **Μηδενικό Γέμισμα (zero-padding)**: Μηδενικό γέμισμα ονομάζεται η προσθήκη εικονοστοιχείων γύρω από την αρχική εικόνα, τα οποία έχουν την τιμή 0. Αυτό συμβαίνει, ώστε να υπάρχει έλεγχος του μεγέθους του χάρτη χαρακτηριστικών που εξάγεται, γιατί αλλιώς θα υπήρχε πάντα μείωση των διαστάσεων. Για παράδειγμα εάν είχαμε έναν πίνακα 5×5 και ένα φίλτρο 3×3, τότε ο χάρτης που θα προέκυπτε θα ήταν αναγκαστικά 3×3, καθώς το φίλτρο δε θα μπορούσε να τοποθετηθεί στα άκρα της εισόδου. Αντίθετα, γεμίζοντας το περίγραμμα με ένα επιπλέον εικονοστοιχείο, ο τελικός χάρτης έχει μέγεθος 5×5 .

4.3.1.2 Στρώμα υποδειγματοληψίας Το στρώμα υποδειγματοληψίας τοποθετείται συνήθως έπειτα από ένα συνελικτικό στρώμα, εξυπηρετώντας τον σκοπό της μείωσης θορύβου που μπορεί να προκληθεί από μικρές διαφοροποιήσεις της εικόνας, αλλά και μείωση των παραμέτρων. Αυτό το καταφέρνει συμπιέζοντας τα δεδομένα των νευρώνων που βρίσκονται στο ίδιο υποδεκτικό πεδίο. Δηλαδή, για μία είσοδο $K \times K$ και ένα φίλτρο $n \times n$, ο χάρτης που προκύπτει μετά την υποδειγματοληψία έχει μέγεθος

$\frac{K}{n} \times \frac{K}{n}$. Υπάρχουν διάφορες παραλλαγές του συγκεκριμένου στρώματος όπως η στοχαστική υποδειγματοληψία (stochastic pooling) [36], η χωρική πυραμιδική υποδειγματοληψία (spatial pyramid pooling) [37] και η υποδειγματοληψία παραμόρφωσης (def-pooling) [38]. Παρ' όλα αυτά, υπάρχουν δύο πιο βασικοί τρόποι για να γίνει αυτή η συμπίεση, που χρησιμοποιούνται λόγω της απλότητας και των καλών αποτελεσμάτων τους. Ο πρώτος και απλούστερος τρόπος, είναι να εξαχθεί η μέση τιμή από αυτούς τους νευρώνες και για αυτό ονομάζεται *υποδειγματοληψία μέσης τιμής* (average pooling), ενώ η έξοδος δίνεται από τη σχέση:

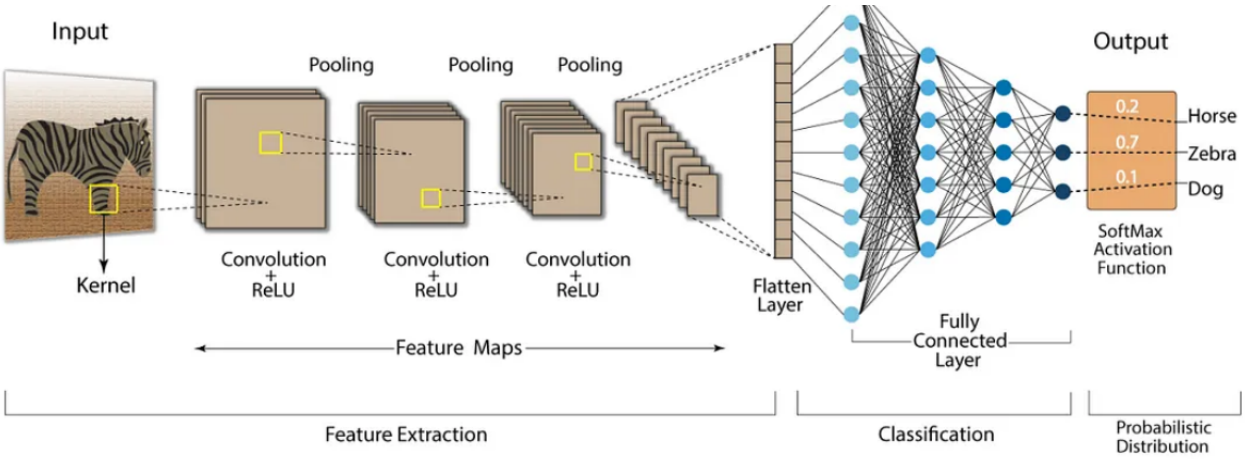
$$\text{Output}(i,j) = \frac{1}{k^2} \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} X_{ik+m,jk+n} \tag{4.3.3}$$

Ο δεύτερος τρόπος, ο οποίος χρησιμοποιείται πιο συχνά επειδή όπως έδειξαν και οι συγγραφείς στο [39], καταφέρνει να προσφέρει πιο αξιόπιστα χαρακτηριστικά, είναι η *υποδειγματοληψία μέγιστης τιμής* (max-pooling). Με αυτόν τον τρόπο, δεν εξάγεται η μέση τιμή των νευρώνων, αλλά η μέγιστη τιμή που υπάρχει στο εκάστοτε υποδεκτικό πεδίο σύμφωνα με τη σχέση:

$$\text{Output}(i,j) = \max_{0 \le m,n \le k-1} x_{ik+m,jk+n} \tag{4.3.4}$$

Σχήμα 4.11: Υποδειγματοληψία μέγιστης τιμής με φίλτρο 2x2 και βήμα 2

4.3.1.3 Πλήρως συνδεδεμένο στρώμα Σε αυτό το στρώμα, όλοι οι νευρώνες έχουν "επικοινωνία" με όλους τους νευρώνες τόσο του προηγούμενο όσο και του επόμενου στρώματος, λειτουργώντας ως ένας κεντρικό σημείο όπου συσσωρεύεται πληροφορία. Σε αυτό το σημείου του δικτύου, τα τοπικά χαρακτηριστικά που εξάγονται από τα προηγούμενα στρώματα, αξιοποιούνται ενιαία ώστε να δημιουργηθεί η τελική κατηγοριοποίηση. Επιπλέον, το πλήρως συνδεδεμένο στρώμα, κάνει χρήση μη γραμμικών συναρτήσεων ενεργοποίησης, με πιο διαδεδομένη να είναι η Softmax προκειμένου να μαθαίνει πιο πολύπλοκα μοτίβα και σχέσεις, σε αντίθεση με τα υπόλοιπα στρώματα που παρουσιάστηκαν, τα οποία χρησιμοποιούν συνήθως τη συνάρτηση ReLU.



Σχήμα 4.12: [40] Ένα παράδειγμα ολόκληρου συνελικτικού νευρωνικού δικτύου

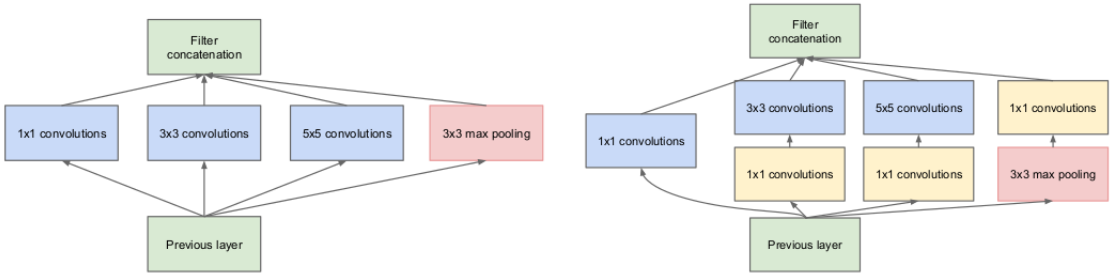
4.3.1.4 Χρήσιμες έννοιες

- **Batch Normalization**: Είναι μια ευρέως διαδεδομένη τεχνική στη βαθιά μάθηση που χρησιμοποιείται για τη σταθεροποίηση και επιτάχυνση της διαδικασίας εκπαίδευσης των νευρωνικών δικτύων. Αποτελεί ένα ακόμα στρώμα του νευρωνικού δικτύου, όπου οι είσοδοι του είναι οι έξοδοι του προηγούμενου στρώματος και οι έξοδοι του είναι οι κανονικοποιημένες είσοδοι. Οι κανονικοποιημένες έξοδοι έχουν μέση τιμή μηδέν και διακύμανση ίση με ένα. Το batch (παρτίδα) αναφέρεται στον τρόπο εισαγωγής δεδομένων στο στρώμα αυτό, που γίνεται με μικρές παρτίδες δεδομένων ανά χρονική στιγμή.
- **Overfitting**: Το overfitting συμβαίνει όταν ένα μοντέλο αποτυγχάνει να μάθει ουσιαστικές πληροφορίες και απομνημονεύει όλες τις λεπτομέρειες και τον θόρυβο από τα δεδομένα εκπαίδευσης. Αυτό έχει ως αποτέλεσμα το μοντέλο να επιτυγχάνει υψηλά ποσοστά στο σύνολο της εκπαίδευσης αλλά να αποτυγχάνει να γενικεύσει αυτά που έχει μάθει και να αποτυγχάνει στο σύνολο δεδομένων ελέγχου. Οι πιθανότητες να συμβεί κάτι τέτοιο αυξάνονται όταν ένα μοντέλο είναι πολύ περίπλοκο και έχει υπερβολικά πολλές παραμέτρους σε σχέση με τα δεδομένα εκπαίδευσης που του παρέχονται. Συχνές τακτικές για να αποφευχθεί το overfitting αποτελούν το regularization, dropout ή, η χρήση περισσότερων δεδομένων.
- **Dropout**: Είναι μία τεχνική που χρησιμοποιείται στα νευρωνικά δίκτυα με το σκοπό να καθυστερήσει ή να αποφευχθεί το overfitting, όπως αναφέρθηκε παραπάνω. Ο τρόπος που δουλεύει είναι διαγράφοντας νευρώνες με τυχαίο τρόπο κατά την εκπαίδευση. Αναλυτικότερα, σε κάθε βήμα της εκπαίδευσης (ένα πέρασμα προς τα εμπρός και ένα πέρασμα προς τα πίσω), η τιμή από ένα σύνολο τυχαίων νευρώνων ορίζεται ως μηδέν. ”Διαγράφοντας” νευρώνες με αυτόν τον τρόπο, το δίκτυο αναγκάζεται να μην στηρίζεται πολύ σε κάποιους συγκεκριμένους, αλλά να μάθει να διανέμει καλύτερα τη πληροφορία σε μεγάλο αριθμό. Είναι μια μέθοδος που έχει παρουσιάσει μεγάλη επιτυχία όσον αφορά την ικανότητα των μοντέλων να γενικεύουν νέα δεδομένα και χρησιμοποιείται εκτενώς στα συνελικτικά δίκτυα.
- **Data augmentation**: Αποτελεί όρο ”ομπρέλα” που περιγράφει διάφορες τεχνικές που αξιοποιούνται για την βελτίωση των δεδομένων. Συμβαίνει κατά την προεπεξεργασία, με σκοπό να αυξηθεί

η ποικιλομορφία των δεδομένων του συνόλου, χωρίς να είναι απαραίτητη η εισαγωγή νέων. Συχνοί τρόπο που επιστρατεύονται για αυτό το σκοπό είναι διάφορες μετατροπές των εικόνων, όπως η περιστροφή, η αλλαγή μεγέθους, η προσθήκη θορύβου. Με αυτόν τον τρόπο το μέγεθος του συνόλου δεδομένων αυξάνεται, ενώ παράλληλα το μοντέλο ”βλέπει” τις εικόνες και με διαφορετικό τρόπο, βελτιώνοντας την διαδικασία εκμάθησης. Είναι ιδιαίτερα χρήσιμο για σύνολα δεδομένων που περιορισμένο αριθμό δειγμάτων και αποτελεί μια από τις πιο συχνές μεθόδους για αποφυγή του overfitting .

4.3.1.5 Αρχιτεκτονικές Συνελικτικών Δικτύων Παρακάτω θα παρουσιαστούν ορισμένες αρχιτεκτονικές από μοντέλα με μεγάλο αντίκτυπο στην πρόοδο των συνελικτικών δικτύων και της τεχνητής νοημοσύνης. Τα μοντέλα που θα παρουσιαστούν αποτελούν τα βασικά CNN που συναντήθηκαν ή χρησιμοποιήθηκαν και στα πλαίσια αυτής της εργασίας με σκοπό την εξαγωγή χρήσιμων χαρακτηριστικών από τα βίντεο με άτομα που εκτελούν νοηματική.

- **GoogLeNet**: Αυτό το μοντέλο [41] προτάθηκε από ερευνητές που εκπροσωπούσαν τη Google στον διαγωνισμό ILSVRC14, όπου κέρδισε τη πρώτη θέση με ποσοστό 6.67% στη κατηγορία top-5. Υπενθυμίζεται πως ο διαγωνισμός αυτός, είναι διαγωνισμός αναγνώρισης και κατηγοριοποίησης εικόνων, χρησιμοποιώντας το σύνολο δεδομένων Imagenet[27]. Η κατηγορία top-5 είναι το αποτέλεσμα από την σύγκριση των πέντε πρώτων προβλέψεων του μοντέλου με τις αληθινές ετικέτες. Η αρχιτεκτονική του μοντέλου αποτελείται κλασικά από μία αλληλουχία συνελικτικών στρωμάτων και στρωμάτων υποδειγματοληψίας, με ειδοποιό διαφορά, ένα καινούριο μπλοκ από αυτά τα στρώματα που παρουσιάζεται για πρώτη φορά, που το ονομάζουν ”Inception module”. Το δεύτερο καινοτόμο στοιχείο του μοντέλου, είναι η χρήση πλευρικών ταξινομητών, τα οποία είναι μικρού μεγέθους συνελικτικά δίκτυα τα οποία χρησιμοποιούνται ως δευτερεύουσες πηγές εκπαίδευσης του βασικού μοντέλου. Παρακάτω φαίνονται αναλυτικά οι δομές αυτών των επιμέρους κομματιών του μοντέλου καθώς και ολόκληρη η αρχιτεκτονική.



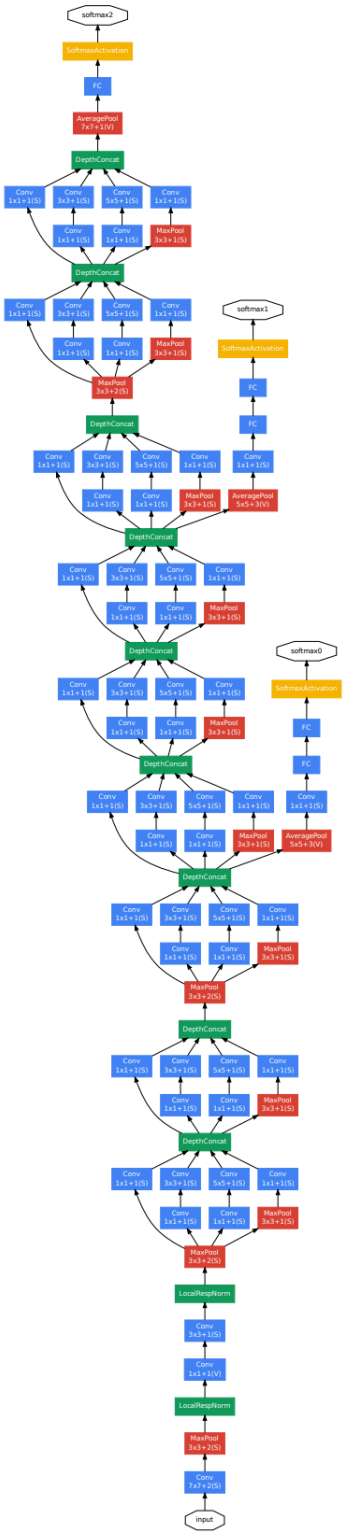
(a) Inception module, naïve version (b) Inception module with dimension reductions

Σχήμα 4.13: [41] Η δομή του *Inception module*

Σχήμα 4.14: Η δομή ενός πλευρικού ταξινομητή

Είδος στρώματος	Χάρτης Χαρακτηριστικών	Μέγεθος Φίλτρου/Βήμα	Αριθμός Παραμέτρων
Συνέλιξη	112×112×64	7×7/2	2.7K
Μέγιστη Υποδειγματοληψία	56×56×64	3×3/2	
Συνέλιξη	56×56×192	3×3/1	112K
Μέγιστη Υποδειγματοληψία	28×28×192	3×3/2	
Inception	28×28×256		159K
Inception	28×28×480		380K
Μέγιστη Υποδειγματοληψία	14×14×480	3×3/2	
Inception	14×14×512		364K
Inception	14×14×512		437K
Inception	14×14×512		463K
Inception	14×14×528		580K
Inception	14×14×832		840K
Μέγιστη Υποδειγματοληψία	7×7×832	3×3/2	
Inception	7×7×832		1072K
Inception	7×7×1024		1388K
Υποδειγματοληψία Μέσης Τιμής	1×1×1024	7×7/1	
Dropout (40%)	1×1×1024		
Πλήρως συνδεδεμένο	1×1×1000		1000K
Softmax	1×1×1000		

Σχήμα 4.15: Η αρχιτεκτονική του δικτύου GoogLeNet. Στα σημεία όπου το μπλοκ Inception είναι γραμμένο με έντονη γραμματοσειρά και υπογραμμισμένο, είναι τα σημεία όπου γίνεται χρήση πλευρικού ταξινομητή.

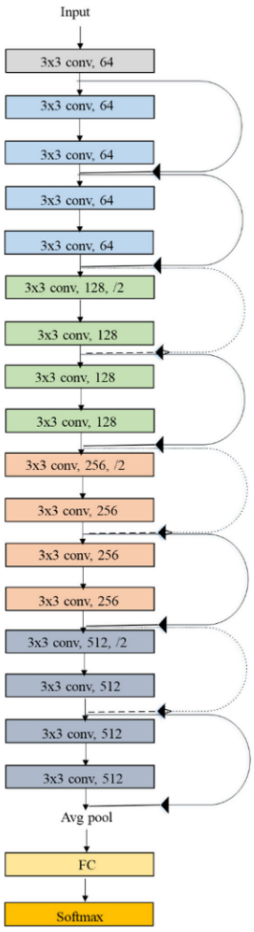


Σχήμα 4.16: [41] Η αναλυτική δομή του δικτύου GoogLeNet

- **Residual Network (ResNet)**: Το δίκτυο ResNet [42] είναι ένα πάρα πολύ διαδεδομένο δίκτυο στον τομέα της αναγνώρισης και μετάφρασης νοηματικής γλώσσας και χρησιμοποιείται ως βασικό cnn για την εξαγωγή χαρακτηριστικών. Ένας λόγος που είναι τόσο δημοφιλές, είναι το ότι έχει βγει σε διάφορες ”εκδόσεις” 18, 34, 50, 101 και 152, οι οποίες αφορούν το πόσα στρώματα έχει

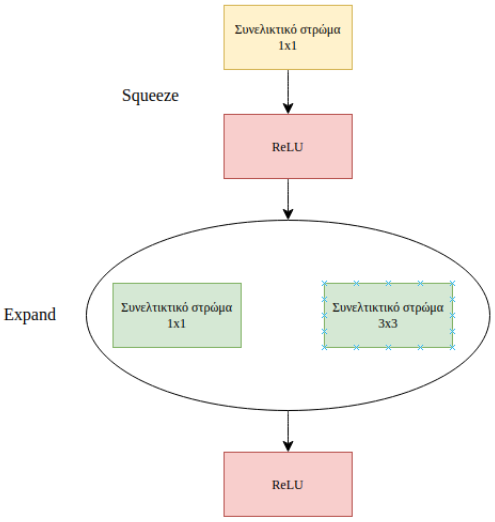
το δίκτυο και πετυχαίνει ιδιαίτερα καλές αποδόσεις με καλή αναλογία υπολογιστικού κόστους-αποτελεσμάτων. Αυτό σημαίνει πως ανάλογα με τις υπολογιστικές απαιτήσεις της διεργασίας προς επίλυση, μπορεί να επιλεχθεί η καταλληλότερη έκδοση. Η επιτυχία του δικτύου αποδίδεται στο καινοτόμο μπλοκ που εισάγει, το *Residual Network Block* το οποίο επιτρέπει τη μεταφορά παραμέτρων μεταξύ στρωμάτων, χωρίς να περιορίζεται απαραίτητα στο αμέσως επόμενο στρώμα. Σκοπός είναι η πιο αποτελεσματική διάδοση πληροφορίας μεταξύ των στρωμάτων, ιδίως κατά την οπισθοδιάδοση, και η ελάττωση του φαινομένου vanishing gradiens. Το μοντέλο ήταν ο νικητής του ILSVRC16 και πέτυχε 4.49% (με το resNet152) στη κατηγορία top-5 .Για το σκοπό της αναγνώρισης νοηματικής γλώσσας, έχει επικρατήσει το Resnet18, το οποίο αποτελείται από:

- 1 στρώμα συνέλιξης που ακολουθείται από batch normalization, ReLU και μέγιστη υποδειγματοληψία
- 4 residual blocks με διαφορετικό αριθμό φίλτρων (64, 128, 256, 512) και βήμα 2
- 1 στρώμα υποδειγματοληψίας μέσης τιμής
- 1 πλήρως συνδεδεμένο στρώμα
- Τελική έξοδος μέσω συνάρτησης ενεργοποίησης Softmax



Σχήμα 4.17: [43] Η δομή του δικτύου ResNet18

- **SqueezeNet**: Η αρχιτεκτονική του SqueezeNet [44] προτάθηκε με κύριο στόχο την μείωση των παραμέτρων και των υπολογιστικών πόρων που απαιτούνται προκειμένου να επιτευχθούν πολύ καλές αποδόσεις. Όπως και στις προηγούμενες περιπτώσεις, οι συγγραφείς χρησιμοποιούν ένα καινούριο μπλοκ το οποίο αυτή τη φορά ονομάζεται *Fire*. Η λογική πίσω από την σχεδίαση αυτού του μπλοκ είναι να μειωθούν οι παράμετροι μέσω τριών βασικών τεχνικών. Συγκεκριμένα γίνεται αντικατάσταση φίλτρων 3×3 με 1×1 , ενώ μειώνονται και τα κανάλια εισόδου. Για την υλοποίηση, λοιπόν (Σχ. 4.18), χρησιμοποιείται ένα συνελικτικό στρώμα 1×1 και στη συνέχεια ένα στρώμα "επέκτασης" που αποτελείται από φίλτρα 1×1 και 3×3. Με αυτόν τον τρόπο αποσκοπούν στην μείωση υπολογισμών αλλά στη διατήρηση συσχετίσεων μεταξύ κοντινών νευρώνων, που επιτρέπει στο μοντέλο να μάθει περίπλοκα χαρακτηριστικά. Η εκπαίδευση του μοντέλου μετράται με βάση το σύνολο δεδομένων ImageNet, όπου πετυχαίνει 80.3% στη κατηγορία top-5, ισάξια με το AlexNet, με τελικό μέγεθος όμως 50 φορές μικρότερο από αυτό του AlexNet. Επίσης οι συγγραφείς παρουσιάζουν και τα αποτελέσματα με κάποιες τεχνικές συμπίεσης στα οποία φαίνεται πως το μέγεθος μπορεί να πέσει μέχρι και 510 φορές μικρότερο από αυτό του AlexNet, χωρίς απόκλιση στα αποτελέσματα.



Σχήμα 4.18: Ένα μπλοκ *Fire*

- **ShuffleNetv2**: Το ShuffleNetv2 [45] αποτελεί μια τροποποίηση του ShuffleNet [46], το οποίο είναι και το αρχικό μοντέλο. Έχει στόχο τη δημιουργία ενός ακόμα πιο αποδοτικού δικτύου, η επίδοση του οποίου δε καθορίζεται μόνο με μείωση υπολογιστικής ισχύος (FLOPS) αλλά και ταχύτητα, δύο χαρακτηριστικά που οι συγγραφείς τονίζουν πως δεν ταυτίζονται. Για να το επιτύχουν αυτό, οι συγγραφείς ορίζουν και ακολουθούν τέσσερις βασικές αρχές. Η πρώτη είναι πως όταν το μέγεθος καναλιών εισόδου και εξόδου παραμένει το ίδιο, ελαχιστοποιείται το κόστος προσπέλασης μνήμης. Η δεύτερη είναι πως η χρήση ομαδικών συνελίξεων σε αλόγιστο βαθμό, επηρεάζει και πάλι αρνητικά το κόστος προσπέλασης μνήμης. Οι ομαδικές συνελίξεις είναι μία τεχνική που χρησιμοποιείται γιατί μειώνει σημαντικά την υπολογιστική ισχύ που απαιτείται για τη συνέλιξη. Αυτό επιτυγχάνεται χωρίζοντας τα κανάλια σε ομάδες και πραγματοποιώντας σε κάθε ομάδα συνέλιξη με ορισμένα από τα φίλτρα. Στο τέλος, οι έξοδοι από κάθε ομάδα ενώνονται δημιουργώντας την τελική έξοδο. Η τρίτη αρχή είναι πως η διάσπαση διαφόρων λειτουργιών σε πολλές μικρότερες,

Είδος στρώματος	Χάρτης Χαρακτηριστικών	Μέγεθος Φίλτρου/Βήμα	Αριθμός Παραμέτρων
Συνέλιξη	111x111x96	7×7/2	14,208
Μέγιστη Υποδειγματοληψία	55x55x96	3×3/2	
Fire	55x55x128	3×3/1	11,920
Fire	55x55x128	3×3/2	12,432
Fire	55x55x256		45,344
Μέγιστη Υποδειγματοληψία	27x27x256	3×3/2	
Fire	27x27x256		49,440
Fire	27x27x384		104,880
Fire	27x27x384		111,024
Fire	27x27x512		188,992
Μέγιστη Υποδειγματοληψία	13x12x512	3×3/2	
Fire	13x13x512		197,184
Dropout	13x13x512		
Συνέλιξη	13x13x1000		513,000
Υποδειγματοληψία Μέσης Τιμής	1x1x1000	7×7/1	
Softmax	1×1×1000		

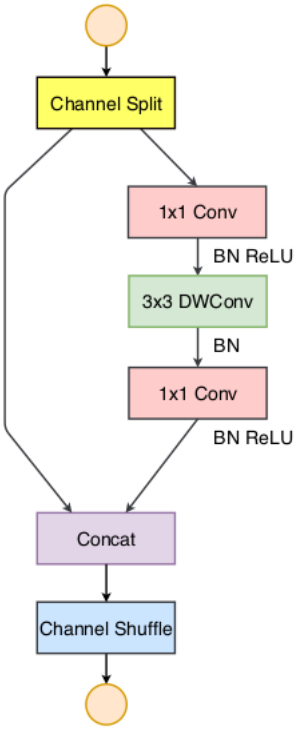
Σχήμα 4.19: Η αρχιτεκτονική του δικτύου SqueezeNet

μειώνει την δυνατότητα παραλληλισμού τους. Τέλος, είναι υπενθύμιση πως οι στοιχειώδεις πράξεις όπως η ReLU δεν είναι αμελητέες, καθώς αυξάνουν δυσανάλογα πολύ τον χρόνο προσπέλασης μνήμης σε σχέση με τη μικρή υπολογιστική ισχύ που απαιτούν.

Βασισμένοι σε αυτές τις αρχές, δομούν ένα καινούριο μπλοκ. Αυτό όπως φαίνεται στο σχήμα 4.20, αρχικά διασπά την είσοδο με βάση τα κανάλια σε δύο ίσα κομμάτια. Το ένα κομμάτι υπόκειται σε συνέλιξη 1×1, έπειτα σε συνέλιξη κατά βάθος 3×3 (κάθε κανάλι συνελίσσεται με ένα ξεχωριστό φίλτρο) και περνάει από ακόμα μία συνέλιξη 1×1. Τέλος, τα δύο κομμάτια ενώνονται και γίνεται κάτι που ονομάζεται *channel shuffle*. Αυτή η τεχνική γίνεται συχνά μετά από χρήση ομαδικών συνελίξεων, ως αντιμετώπιση του προβλήματος πως κάθε ομάδα έχει ”αλληλεπιδράσει” μόνο με ορισμένα κανάλια εισόδου, το οποίο μπορεί να μειώσει την απόδοση. Κατά το channel shuffle, η πληροφορία των καναλιών ξαναμοιράζεται τυχαία ώστε να μειωθεί αυτό το πρόβλημα. Συνολικά η αρχιτεκτονική του ShuffleNetv2 φαίνεται στο σχήμα 4.21.

4.4 Long Short-Term Memory

Το δίκτυο Long Short-Term Memory (LSTM) είναι ένα είδος αναδρομικού νευρωνικού δικτύου (Reccurent Neural Network - RNN) το οποίο αποσκοπεί στην βελτίωση των παραδοσιακών RNN, ιδιαίτερα στον τομέα της εκμάθησης μακροπρόθεσμων συσχετίσεων σε σειριακά δεδομένα. Προτάθηκε πρώτη φορά το 1997 [47] με σκοπό την εξάλειψη δύο βασικών προβλημάτων που εμφάνιζαν τα RNN, λόγω του τρόπου



Σχήμα 4.20: Το δομικό μπλοκ του δικτύου ShuffleNetv2

Είδος στρώματος	Χάρτης Χαρακτηριστικών	Μέγεθος Φίλτρου/Βήμα	Επανάληψη	Έξοδος
Εισοδος	224 x 224			3
Συνέλιξη	112×112	3×3/2	1	24
Μέγιστη Υποδειγματοληψία	56×56	3×3/2	1	24
Stage	28×28 28×28	/2 /1	1 3	116
Stage	14×14 14×14	/2 /1	1 7	232
Stage	7×7 7×7	/2 /1	1 3	464
Συνέλιξη	7×7	1×1/1	1	1024
Υποδειγματοληψία Μέσης Τιμής	1×1	7×7		
Πλήρως συνδεδεμένο	1×1×1000			
Softmax	1×1×1000			

Σχήμα 4.21: Η αρχιτεκτονική του δικτύου ShuffleNetv2. Κάθε στρώμα stage αποτελεί ένα δομικό μπλοκ, επαναλαμβανόμενο όσες φορές φαίνεται στη στήλη 'Επαναλήψεις'. Για παράδειγμα το πρώτο στρώμα stage περιέχει 1 μπλοκ με βήμα 2 και 3 ακόμα με βήμα 1.

που εκπαιδεύονται μέσω του κανόνα Back-Propagation Through Time και παραλλαγών του. Συγκεκριμένα τα προβλήματα είναι:

- Exploding Gradients: Κατά την εκμάθηση, τα σήματα λάθους που καθορίζουν τις αλλαγές των βαρών του μοντέλου, μεγαλώνουν πολύ γρήγορα. Αυτό έχει ως αποτέλεσμα να αλλάζουν με μεγάλο ρυθμό τα βάρη και να προκαλούνται αστάθειες που επηρεάζουν αρνητικά την διαδικασία εκπαίδευσης.

- Vanishing Gradients: Το αντίθετο που μπορεί να συμβεί, είναι τα σήματα λάθους να γίνουν υπερβολικά μικρά κατά τη μεταφορά με αποτέλεσμα να μην μπορούν να παρέχουν αρκετή πληροφορία ώστε να μάθει το δίκτυο. Έτσι η διαδικασία εκμάθησης μπορεί να γίνει πολύ αργή ή και αδύνατη.

Αυτό που καθιστά το LSTM ένα πολύ αποδοτικό δίκτυο, είναι η ιδιαίτερη αρχιτεκτονική του η οποία χρησιμοποιεί για πρώτη φορά μία δομή που την ονομάζουν *memory cell* το οποίο αποτελείται από πύλες εισόδου (input gates), πύλες λήθης (forget gates) και πύλες εξόδου (output gates). Αυτές οι πύλες είναι υπεύθυνες για τη διαχείριση των δεδομένων στο δίκτυο. Η πύλη εισόδου είναι υπεύθυνη για τα δεδομένα που εισέρχονται, η πύλη εξόδου ελέγχει τα δεδομένα που εξέρχονται από το μοντέλο, ενώ η πύλη λήθης ανάλογα με την τιμή που έχει, αποφασίζει για πόσα χρονικά βήματα θα ”θυμάται” την τιμή της προηγούμενης κατάστασης.

Το 2005 οι Graves και Schmidhuber δημοσίευσαν ένα άρθρο στο οποίο χρησιμοποιούσαν μία δομή που περιλάμβανε δύο LSTM. Το ένα εκπαιδευόταν με καταστάσεις από το παρελθόν προς το παρόν, ενώ το άλλο με καταστάσεις από το μέλλον προς το παρόν. Αυτή η αμφίδρομη δομή, η οποία ονομάστηκε **Bidirectional LSTM (BiLSTM)** [48], επιτρέπει στο μοντέλο να αξιοποιεί πληροφορία τόσο από την αρχή, όσο και από το τέλος μιας σειριακής εισόδου, καθιστώντας τα ιδιαίτερα αποδοτικά σε καταστάσεις όπου ολόκληρη η ακολουθία είναι σημαντική. Έκτοτε τα BiLSTM έχουν αποτελέσει ακρογωνιαίο λίθος στην ανάπτυξη αρχιτεκτονικών σε τομείς όπως η επεξεργασία φυσικής γλώσσας, αναγνώριση ομιλίας και συναισθήματος αλλά και γενικά σε διεργασίες που περιλαμβάνουν επεξεργασία διαφόρων ακολουθιών.

4.5 Connectionist Temporal Classification

Όπως αναφέρθηκε παραπάνω, τα RNN διαδραμάτισαν μεγάλο ρόλο στον τομέα της εκμάθησης ακολουθιών. Παρ’όλα αυτά, για ένα μεγάλο διάστημα, εξακολουθούσαν να πάσχουν όταν καλούνταν να πραγματοποιήσουν προβλέψεις σχετικά με μια σειρά μη κατατετμημένων ετικετών. Εάν τα δεδομένα, λοιπόν, δεν είναι χωρισμένα κατάλληλα, υπάρχει μεγάλο αρνητικό αντίκτυπο στις προβλέψεις των μοντέλων. Αυτόν τον τομέα προσπάθησαν να θεραπεύσουν οι συγγραφείς του [49], δημιουργώντας το δίκτυο Connectionist Temporal Classification (CTC). Συγκεκριμένα το κομμάτι μεγαλύτερου ενδιαφέροντος για τη παρούσα εργασία, είναι η συνάρτηση απώλειας που υλοποιούν και χρησιμοποιούν προκειμένου να εκπαιδεύσουν το δίκτυο, που ονομάζεται *απώλεια CTC* (CTC loss). Σκοπός της απώλειας CTC είναι να ευθυγραμμίσει χρονικά την είσοδο με τις ετικέτες-στόχους, όταν δεν είναι γνωστή εκ των προτέρων. Για να το επιτύχει αυτό, εισάγει τον ’κενό’ χαρακτήρα στο υπάρχων λεξιλόγιο προκειμένου να αντιπροσωπεύει δεδομένα χωρίς ετικέτα ενώ παράλληλα επιτρέπει στο μοντέλο να προσπεράσει ή να επαναλάβει εισόδους. Έτσι δημιουργείται το καινούριο σύνολο ετικετών-στόχων $\mathbf{G}' = \mathbf{G} \cup \{\text{blank}\}$, όπου $\mathbf{G}$, είναι το αρχικό σύνολο ετικετών. Αφού κάνει τις αρχικές προβλέψεις, χρησιμοποιεί μία μέθοδο many-to-one προκειμένου να αφαιρέσει τις διπλότυπες συνεχόμενες εμφανίσεις ετικετών-στόχων. πχ. $\mathcal{B}(-cc---cccddddd-) = \mathcal{B}(-c-cd-) = ccd$.

Τέλος, για να βρεθεί η απώλεια CTC, υπολογίζεται το άθροισμα των πιθανοτήτων όλων των εφικτών ακολουθιών ετικετών-στόχων,οι οποίες αντιστοιχούν στις πιθανές ευθυγραμμίσεις της εισόδου με τις ετικέτες-στόχους, σύμφωνα με τη σχέση [16]:

$$\mathcal{L}_{CTC} = -\log p(\boldsymbol{l} \mid \boldsymbol{X}) = -\log(\sum_{\pi \in \mathcal{B}^{-1}(\boldsymbol{l})} p(\pi \mid \boldsymbol{X})) \tag{4.5.1}$$

Όπου $\mathbf{X}$ είναι μια ακολουθία T εισόδων, $\boldsymbol{l} \in \mathbf{G}^{\leq T}$ είναι μια ακολουθία N ετικετών-στόχων και $\pi \in \mathbf{G}'^{T}$ η ευθυγραμμισμένη ακολουθία από εισόδους με ετικέτες-στόχους.

Κεφάλαιο 5ο: Μεθοδολογία

Σε αυτό το κομμάτι, θα αναλυθεί ολόκληρη η διαδικασία που ακολουθήθηκε, από την επιλογή μοντέλου μέχρι την εξαγωγή αποτελεσμάτων. Αρχικά, θα απαντηθεί η ερώτηση, γιατί να γίνει μελέτη μοντέλου CSLR και όχι SLT, αφού η μετάφραση είναι ο τελικός στόχος. Έπειτα, θα περιγραφούν οι λόγοι και η μεθοδολογία με την οποία επιλέχθηκε το μοντέλο που χρησιμοποιείται και θα αναλυθεί το θεωρητικό υπόβαθρο του συγκεκριμένου μοντέλου. Στη συνέχεια θα αναπτυχθούν κάποια εργαλεία που χρησιμοποιήθηκαν και θα αναλυθεί η διαδικασία ρύθμισης του περιβάλλοντος που χρειάζεται για να τρέξει το μοντέλο. Επιπροσθέτως, θα εξεταστεί η διαδικασία προεπεξεργασίας των δεδομένων, και οι παράμετροι που χρησιμοποιήθηκαν για την εκπαίδευση. Τέλος, θα εξηγηθεί γιατί επιλέχθηκαν τα συγκεκριμένα στοιχεία για την εξαγωγή των αποτελεσμάτων, καθώς και οι λόγοι που οδήγησαν στην προτίμησή τους έναντι άλλων πιθανών επιλογών.

5.1 Συνεχής αναγνώριση έναντι μετάφρασης

Ο πρώτος και πιο σημαντικός λόγος για τον οποίο επιλέχθηκε η μελέτη του τομέα συνεχής αναγνώρισης έναντι της μετάφρασης νοηματικής γλώσσας, είναι γιατί όπως αναφέρθηκε και στο δεύτερο κεφάλαιο, πολλά μοντέλα SLT, είναι μοντέλα CSLR στα οποία προστίθεται ένα επιπλέον κομμάτι μετάφρασης. Αυτό σημαίνει πως η έρευνα και βελτίωση μοντέλων CSLR έχει άμεσο αντίκτυπο και στα μοντέλα SLT. Ο δεύτερος λόγος είναι πως ο τομέας της μετάφρασης ξεκίνησε στηριζόμενος στην προϋπάρχουσα δουλειά πάνω στην αναγνώριση, οπότε η εισαγωγή σε αυτόν τον ερευνητικό τομέα, είναι καλύτερο να γίνει από την τελευταία. Τέλος, τα μοντέλα SLT συχνά εισάγουν πρόσθετη πολυπλοκότητα, η οποία, λόγω των περιορισμών του διαθέσιμου υλισμικού (hardware), θα καθιστούσε τη λειτουργία τους εξαιρετικά δύσκολη.

5.2 Επιλογή Μοντέλου

Αρχικός σκοπός ήταν να εξεταστούν τα state-of-the-art μοντέλα που είχαν διαθέσιμο κώδικα τους, σε πλατφόρμες όπως το GitHub. Για αυτό το σκοπό εξετάστηκαν στην αρχή τα μοντέλα CorrNet+ [17], TwoStreamSLR [21] καθώς και το Towards Online CSLRT [50], λόγω της καινοτομίας του στην ανάπτυξη ενός μοντέλου που τρέχει σε πραγματικό χρόνο. Από αυτά, τα δύο τελευταία παρουσίασαν έντονες δυσκολίες στην αρχική εγκατάσταση όλων των απαραίτητων βιβλιοθηκών λόγω ασυμβατοτήτων. Το CorrNet+, έτρεξε εν μέρει με επιτυχία, αλλά σύντομα έγινε εμφανές πως λόγω ανεπαρκούς υλισμικού (κυρίως κάρτα γραφικών), η εκπαίδευση δεν μπορούσε να πραγματοποιηθεί με τις αρχικές παραμέτρους. Η πλήρης ανάλυση των λόγων θα πραγματοποιηθεί σε αργότερο σημείο στην εργασία, στο κομμάτι των προβλημάτων που αντιμετωπίστηκαν.

Αυτό οδήγησε στη συνειδητοποίηση πως στη βιβλιογραφία, τα μοντέλα που αξιοποιούνται είναι πολύ μεγάλα, με πολλές παραμέτρους, καθιστώντας τα ανεπαρκή για χρήση σε πραγματικές συνθήκες, όπου η λειτουργία σε πραγματικό χρόνο και σε συσκευή με περιορισμένη μνήμη είναι θεμιτή, αν όχι απαραίτητη. Επιπλέον, η ανάγκη για ισχυρές, εξειδικευμένες κάρτες γραφικών (που συναντούνται κυρίως σε ερευνητικά εργαστήρια) για την εκπαίδευση αυτών των μοντέλων, περιορίζει σημαντικά τον αριθμό των ατόμων που μπορούν να συμβάλουν στην εξέλιξη του τομέα, αποκλείοντας όσους διαθέτουν μόνο πιο

προσιτές ή κοινές συσκευές. Ενδεικτικά, κάποιες από τις κάρτες γραφικών που συναντώνται στη βιβλιογραφία είναι GeForce RTX 3090, συστοιχία από 4 GTX 1080Ti, συστοιχία από 8 Nvidia V100 GPUs κ.ά.

Έτσι, ο τελικός στόχος έγινε η τροποποίηση ενός μοντέλου ώστε να λειτουργεί με χαμηλότερες απαιτήσεις, η σύγκρισή του με το αρχικό καθώς και με άλλα μοντέλα αιχμής (state-of-the-art) και η δοκιμή του στο σύνολο δεδομένων GSL. Για να επιτευχθεί ο στόχος, έπρεπε το επιλεγμένο μοντέλο να πληρεί τα παρακάτω κριτήρια:

- Κώδικα που ακολουθεί καλές πρακτικές, ώστε να είναι εφικτή η τροποποίηση του ανάλογα με τις ανάγκες.
- Να χρησιμοποιεί βελτιστοποιήσεις για σωστή διαχείριση της κάρτας γραφικών
- Να είναι σχετικά ”ελαφρύ”, δηλαδή να μην κάνει χρήση περίπλοκων μεθόδων για την βελτίωση της απόδοσης που παράλληλα αυξάνουν δυσανάλογα την περιπλοκότητα και το μέγεθος.

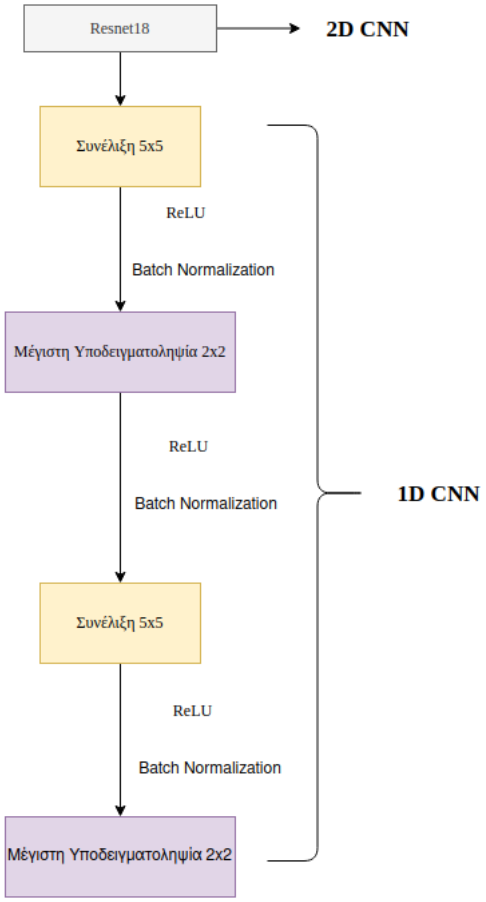
Το μοντέλο που επιλέχθηκε με βάση αυτά τα κριτήρια είναι το VAC_CSLR [16] καθώς ο κώδικας του είναι καλά δομημένος, αποτελεί τη βάση που χρησιμοποίησαν μοντέλα όπως το CorrNet+ που αναφέρθηκε προηγουμένως (και θεωρείται state-of-the-art) και το SEN_SLR [19]. Επιπλέον χρησιμοποιεί βελτιστοποιήσεις, ιδίως στο κομμάτι φόρτωσης δεδομένων, το οποίο είναι ζωτικής σημασίας για τη καλή χρήση της μνήμης της κάρτας γραφικών. Τέλος, είναι ”ελαφρύ” μοντέλο καθώς χρησιμοποιεί μόνο ένα επιπλέον κομμάτι (πέρα από τη συχνή αρχιτεκτονική που αναλύθηκε σε προηγούμενα κεφάλαια) το οποίο αποτελείται από δύο συμπληρωματικές συναρτήσεις απώλειας. Παράλληλα, δεν υστερεί σε μεγάλο βαθμό στην απόδοση, έχοντας μόλις 4% χαμηλότερη απόδοση σε σχέση με το CorrNet+. Παρακάτω θα περιγραφεί αναλυτικά το θεωρητικό υπόβαθρο του VAC_CSLR

5.3 Θεωρητικό Υπόβαθρο Μοντέλου

Το μοντέλο χρησιμοποιεί την αρχιτεκτονική που περιγράφηκε και στη βιβλιογραφική ανασκόπηση, που περιλαμβάνει τον εξαγωγέα χαρακτηριστικών κι ένα κομμάτι ευθυγράμμισης.

Ο εξαγωγέας αποτελείται από ένα 2D CNN και συγκεκριμένα το ResNet18 προεκπαιδευμένο πάνω στο σύνολο δεδομένων ImageNet, το οποίο εξάγει χαρακτηριστικά σε επίπεδο μεμονωμένων καρέ, και ένα 1D CNN το οποίο εξάγει βραχυπρόθεσμες χρονικές συσχετίσεις μεταξύ των χαρακτηριστικών πήρε από το 2D CNN. Το 1D CNN είναι δομημένο σύμφωνα με τη state-of-the-art αρχιτεκτονική K5-P2-K5-P2, όπου K5 είναι ένα στρώμα συνέλιξης με φίλτρο μεγέθους 5×5 και P2 είναι ένα στρώμα μέγιστης υποδειγματοληψίας με φίλτρο μεγέθους 2×2. Ενδιάμεσα σε καθένα από αυτά τα στρώματα, υπάρχει μία συνάρτηση ενεργοποίησης ReLU και ένα στρώμα batch normalization.

Το επόμενο κομμάτι, είναι υπεύθυνο για την ευθυγράμμιση των ετικετών-στόχων. Αναλυτικότερα, χρησιμοποιούν ένα BiLSTM για τις μακροπρόθεσμες χρονικές συσχετίσεις και απώλεια CTC για βελτίωση της ευθυγράμμισης, ακολουθώντας πάλι τη συνηθισμένη αρχιτεκτονική. Επιπλέον όμως, εισάγουν έναν περιορισμό οπτικής ευθυγράμμισης *(Visual Alignment Constraint)* ο οποίος αποτελείται από δύο ακό-



Σχήμα 5.1: Το 1D CNN

μα βοηθητικές συναρτήσεις απώλειας με σκοπό να βελτιώσουν τον εξαγωγέα χαρακτηριστικών. Αυτές είναι:

- **Απώλεια οπτικής ενίσχυσης** (Visual Enhancement-VE loss) → Μια επιπλέον απώλεια CTC η οποία χρησιμοποιεί τις εξόδους από έναν δευτερεύων ταξινομητή, ο οποίος δέχεται ως είσοδο μόνο τα οπτικά χαρακτηριστικά που έχουν εξαχθεί από το δίκτυο 2D+1D, πριν αυτά εισαχθούν στο BiLSTM. Με αυτόν τον τρόπο, προσφέρεται αυξημένη καθοδήγηση στον εξαγωγέα χαρακτηριστικών. Η μαθηματική έκφραση της απώλειας είναι:

$$\mathcal{L}_{VE} = \mathcal{L}_{CTC}^{v} = -\log p(\boldsymbol{l}|\boldsymbol{X};\theta^{v}). \tag{5.3.1}$$

  Όπου $\boldsymbol{X}$ μια ακολουθία από N καρέ, $\boldsymbol{l}$ μια ακολουθία από M γλωσσικές μονάδες και θ^{v} παράμετροι στην έξοδο του εξαγωγέα χαρακτηριστικών.

- **Απώλεια οπτικής ευθυγράμμισης** (Visual Alignment-VA loss) → Αυτή η απώλεια έρχεται ως συμπληρωματική της απώλειας VE, προκειμένου να δημιουργηθούν λανθασμένες ευθυγραμμίσεις μεταξύ της τελευταίας και της βασικής απώλειας CTC. Συγκεκριμένα, υλοποιείται ως *knowledge distillation loss* χρησιμοποιώντας την απόκλιση Kullback-Leibler για να συγκρίνει τις κατανομές των εξόδων $\boldsymbol{Y}$ και $\tilde{\boldsymbol{Y}}$. Τα $\boldsymbol{Y}$ και $\tilde{\boldsymbol{Y}}$ είναι τα προβλεπόμενα logits (μη κανονικοποιημένες έξοδοι-προβλέψεις του δικτύου) και τα προβλεπόμενα logits που αφορούν μόνο τα οπτικά χαρακτηριστι-

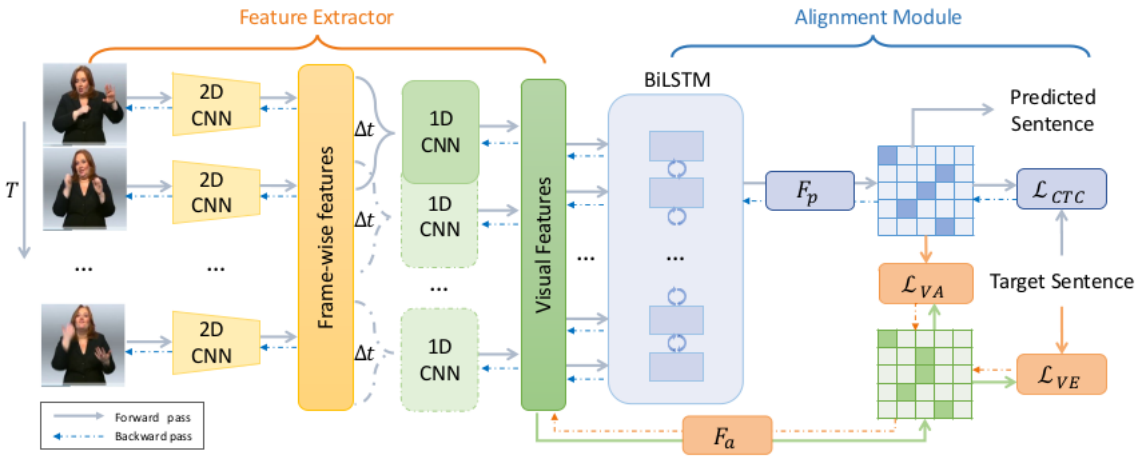
κά, αντίστοιχα. Η μαθηματική σχέση παρέχεται παρακάτω:

$$\mathcal{L}_{VA} = \mathrm{KL}(\mathrm{softmax}(\frac{\boldsymbol{Y}}{\tau}), \mathrm{softmax}(\frac{\tilde{\boldsymbol{Y}}}{\tau})) \tag{5.3.2}$$

Το τ παίρνει την σχετικά μεγάλη τιμή 8, προκειμένου να εξομαλύνει την κατανομή πιθανοτήτων.

Συνολικά, λοιπόν, η συνάρτηση απώλειας ορίζεται ως το άθροισμα όλων των επιμέρους απωλειών ως:

$$\mathcal{L} = \mathcal{L}_{CTC} + \mathcal{L}_{VE} + \alpha\mathcal{L}_{VA}, \quad \alpha = 25 \tag{5.3.3}$$



Σχήμα 5.2: Η αρχιτεκτονική του μοντέλου VAC-CSLR. Ο πρωτεύων ταξινομητής είναι ο F_p, ενώ ο δευτερεύων είναι ο F_a.

Ο λόγος για τον οποίον οι συγγραφείς θεωρούν πως ο εξαγωγέας χαρακτηριστικών χρειάζεται κάποια επιπλέον καθοδήγηση, οφείλεται στον τρόπο που λειτουργεί η απώλεια CTC. Συγκεκριμένα, όπως αναφέρθηκε σε προηγούμενο κεφάλαιο, η απώλεια CTC χρησιμοποιεί τον κενό χαρακτήρα για να συμπληρώσει τις προβλέψεις για χρονικές στιγμές που δεν έχει κατάλληλα δεδομένα. Για αυτό το λόγο, συχνά οι αποκρίσεις των εξόδων τείνουν να σχηματίζουν αιχμές, οι οποίες αποτελούνται από κάποια συγκεκριμένα καρέ που προκαλούν μεγάλη βεβαιότητα για την ετικέτα-στόχο στην οποία αντιστοιχούν, ενώ στα υπόλοιπα ανατίθεται ο κενός χαρακτήρας. Αυτά τα σημαντικά καρέ, μονοπωλούν την προσοχή του δικτύου BiLSTM με αποτέλεσμα να μειώνονται σημαντικά οι είσοδοι/καρέ που ενδιαφέρουν το μοντέλο. Σε συνδυασμό με τα μικρά σύνολα δεδομένων, για τις ανάγκες των δικτύων, αυτό μπορεί να προκαλέσει σημαντικό overfitting. Με αυτόν τον τρόπο καταλήγουν στην υπόθεση πως η βελτίωση του εξαγωγέα χαρακτηριστικών είναι ζωτικής σημασίας για τέτοιου είδους μοντέλα.

Το μοντέλο πέτυχε 22.3% WER στο σύνολο δεδομένων phoenix14, ένα από τα κορυφαία για τη χρονολογία δημοσίευσης, το 2021, ενώ παράλληλα είναι ελαφρύ και γρήγορο σε σχέση με υπόλοιπα, και χωρίς να κάνει χρήση πολυτροπικών δεδομένων.

5.4 Εργαλεία που χρησιμοποιήθηκαν

Η εργασία πραγματοποιήθηκε στο λειτουργικό σύστημα Ubuntu 22.04 LTS καθώς ήταν απαραίτητο για την λειτουργία της βιβλιοθήκης που χρειαζόταν για την λειτουργία απώλειας CTC. Κάποια σημαντικά εργαλεία/λογισμικά που χρησιμοποιήθηκαν και αξίζουν αναφορά είναι:

- **Anaconda**: Το anaconda είναι ένα λογισμικό ανοιχτού κώδικα το οποίο έχει σχεδιαστεί για να διευκολύνει τη διαχείριση πακέτων, τη διαχείριση περιβάλλοντος και την ανάπτυξη προγραμμάτων για τις γλώσσες Python και R. Παρέχει πληθώρα απαραίτητων βιβλιοθηκών και πακέτων, ενώ παρέχει και εργαλεία όπως το Jupyter Notebooks και Spyder, για την διαδραστική ανάλυση δεδομένων. Ένα από τα πιο χρήσιμα χαρακτηριστικά του, είναι η δυνατότητα δημιουργίας και διαχείρισης ανεξάρτητων περιβαλλόντων, εκμηδενίζοντας τον κίνδυνο για ασυμβατότητες μεταξύ βιβλιοθηκών. Όλα αυτά τα χαρακτηριστικά το κάνουν ένα πολύ χρήσιμο εργαλείο για τομείς όπως επεξεργασία δεδομένων και μηχανική μάθηση

- **Git**:Το Git είναι το πιο διαδεδομένο *version control system*, δηλαδή σύστημα που επιτρέπει τον έλεγχο εκδόσεων. Είναι ανοιχτού κώδικα και μπορεί να χειριστεί από μικρά μέχρι μεγάλα προγράμματα με ταχύτητα. Επιτρέπει στους προγραμματιστές να έχουν μία κοινή βάση κώδικα και να παρακολουθούν τις αλλαγές, να τις διαχειρίζονται και να τις αρχειοθετούν κατάλληλα. Χρησιμοποιείται παντού για τη συγγραφή λογισμικού, από μικρές ομάδες ατόμων μέχρι μεγάλες εταιρείες.

- **GitHub**: Το GitHub είναι μία πλατφόρμα βασισμένη στο Git, η οποία προσφέρει έναν χώρο για τους προγραμματιστές για να αποθηκεύουν και να διαχειρίζονται τον κώδικα τους. Προσφέρει μία διαδικτυακή διεπαφή για το git και επιπλέον χαρακτηριστικά που συμβάλλουν στην ομαλή διαχείριση κώδικα και διαχείριση ανθρώπινου δυναμικού όταν δουλεύουν παράλληλα πολλά άτομα πάνω στην ίδια βάση κώδικα. Επίσης το GitHub προσφέρει τόσο δημόσια όσο και ιδιωτικά *repositories*, όπου μπορεί να αποθηκευτεί κώδικας είτε για ιδιωτική ανάπτυξη είτε για δημόσια προβολή, καθιστώντας το μια ιδιαίτερα ελκυστική πλατφόρμα για τη στέγαση προγραμμάτων ανοιχτού κώδικα. Τέλος, ενσωματώνεται με *CI/CD pipelines*, επιτρέποντας την αυτοματοποιημένη δοκιμή και ανάπτυξη εφαρμογών.

- **PyTorch**: Το PyTorch είναι ένα framework ανοιχτού κώδικα για βαθιά μάθηση, που δημιουργήθηκε από τη Facebook. Χρησιμοποιείται εκτενώς για διεργασίες όπως μηχανική όραση, επεξεργασία φυσικής γλώσσας και αναγνώρισης ενεργειών. Παρέχει ένα πολύ εύχρηστο API, δυνατότητες επιτάχυνσης μέσω χρήσης καρτών γραφικών, καθώς και έτοιμα προεκπαιδευμένα μοντέλα για γρήγορη χρήση. Αυτά τα χαρακτηριστικά και επιπλέον η μεγάλη κοινότητα του, το καθιστούν πολύ δημοφιλές, ιδιαίτερα στους ερευνητές αλλά και στους επαγγελματίες.

- **Visual Studio Code**: Το VS Code είναι ένα δωρεάν ανοιχτό λογισμικό, για συγγραφή κώδικα της Microsoft. Έχει σχεδιαστεί για να βοηθάει στη συγγραφή προγραμμάτων σε πληθώρα γλωσσών, συμπεριλαμβανομένων των Python, JavaScript, C++ κ.ά. Είναι πολύ διαδεδομένο λόγω της ταχύτητας του και του οικοσυστήματος του, το οποίο παρέχει πολλές επεκτάσεις που επιτρέπουν την εύκολη προσαρμογή του περιβάλλοντος στις απαιτήσεις του κάθε προγραμματιστή. Μέσω των επεκτάσεων μπορούν επιπλέον να προστεθούν νέες λειτουργίες και να βελτιστοποιηθεί η αποδοτικότητα. Κάποια από τα πιο δημοφιλή χαρακτηριστικά είναι το ενσωματωμένο Git, εργαλεία

που διευκολύνουν την αποσφαλμάτωση (debugging) και το IntelliSense που επιτρέπει στη γρήγορη ολοκλήρωση κώδικα. Συνολικά, ο ελαφρύς του σχεδιασμός και η δυνατότητα επέκτασης και προσαρμοστικότητας, το κάνουν ένα από τα πιο δημοφιλή εργαλεία συγγραφής κώδικα.

5.5 Επιλογή συνόλων δεδομένων

- RWTH-PHOENIX-Weather-2014: Η επιλογή του συγκεκριμένου συνόλου έγινε κυρίως λόγω της εκτενούς χρήσης του στη βιβλιογραφία. Όπως αναφέρθηκε, ένας από τους σκοπούς της συγκεκριμένης εργασίας είναι να παρέχει καινούρια δεδομένα σχετικά με τα θετικά και τα αρνητικά χαρακτηριστικά που επιφέρει η χρήση πιο ”ελαφριών” backbones, καθώς και μειωμένου μεγέθους εικόνων. Προκειμένου αυτά τα δεδομένα να είναι συγκρίσιμα με άλλα μοντέλα που παρουσιάστηκαν στη βιβλιογραφία, χρησιμοποιήθηκε αυτό το σύνολο για να προσφέρει ένα αντικειμενικό μέτρο σύγκρισης.

- Greek Sign Language Dataset: Η επιλογή αυτού του συνόλου έγινε για τρεις λόγους. Πρώτον, στόχευε στην προώθηση της έρευνας σχετικά με την αναγνώριση και μετάφραση της Ελληνικής Νοηματικής Γλώσσας. Δεύτερον, επιδίωκε να δοκιμάσει την απόδοση που θα επέφερε η χρήση ενός διαφορετικού μοντέλου από αυτά που είχαν χρησιμοποιηθεί στην αρχική εργασία που παρουσίασε το σύνολο δεδομένων. Τέλος, το σύνολο αυτό παρουσιάζει σημαντικές διαφορές με το Phoenix14, τόσο στη γλώσσα όσο και σε χαρακτηριστικά όπως ο δραστικά διαφορετικός αριθμός γλωσσικών μονάδων και μοναδικών προτάσεων. Λόγω αυτού, θεωρήθηκε ένα καλό σύνολο προκειμένου να διασταυρωθεί η ορθότητα των αποτελεσμάτων που θα παραχθούν.

5.6 Προεπεξεργασία Δεδομένων

Η προεπεξεργασία των δεδομένων είναι ένα πολύ σημαντικό κομμάτι καθώς συχνά τα δεδομένα πρέπει να καθαριστούν και να τροποποιηθούν ώστε να είναι σε κατάλληλη μορφή για να μπορούν να τροφοδοτηθούν στο δίκτυο. Στη προκειμένη περίπτωση όμως, λόγω κατάλληλης μέριμνας των δημιουργών των συνόλων δεδομένων, δεν χρειαζόντουσαν διορθώσεις. Η προεπεξεργασία που έγινε, λοιπόν, αφορούσε κυρίως την αλλαγή των διαστάσεων των εικόνων και κάποιες τεχνικές με σκοπό την αποφυγή του overfitting. Λόγω κάποιων διαφορών μεταξύ των δύο συνόλων δεδομένων, θα περιγραφούν ξεχωριστά.

5.6.0.1 Προεπεξεργασία RWTH-PHOENIX-Weather-2014 : Η διαδικασία ξεκινάει με την αλλαγή μεγέθους των εικόνων-καρέ. Συγκεκριμένα, τα καρέ από το σύνολο δεδομένων παρέχονται σε μέγεθος 210×260 και μετατρέπονται σε 256×256. Παράλληλα, τα βίντεο χωρίζονται σε φακέλους train,dev,test ακολουθώντας τον διαχωρισμό με πολλούς διερμηνείς (mutlisigner) που παρέχουν τα δεδομένα. Επίσης δημιουργείται ένα αρχείο που περιλαμβάνουν όλες τις ξεχωριστές γλωσσικές μονάδες σε μορφή λεξικού, ένα αρχείο που περιλαμβάνει όλες τις απαραίτητες πληροφορίες του συνόλου και περιέχει ουσιαστικά τις αντιστοιχίσεις βίντεο-προτάσεων ετικετών και τέλος τρία αρχεία που λειτουργούν ως βασική αλήθεια (groundtruth) για την αξιολόγηση των προβλέψεων. Αυτό αποτελεί το πρώτο κομμάτι επεξεργασίας που εκτελείται πριν την έναρξη της εκπαίδευσης. Το δεύτερο κομμάτι περιλαμβάνει την διαδικασία που εκτελείται για data augmentation, το οποίο συμβαίνει κατά την φόρτωση των δεδομένων στον DataLoader,

ώστε να μην απαιτείται επιπλέον αποθηκευτικός χώρος στον δίσκο. Οι τροποποιήσεις που υλοποιούνται είναι:

- **Random Crop**: Στο αρχικό μοντέλο γίνεται τυχαία περικοπή ώστε το τελικό μέγεθος της εικόνας να είναι 224×224. Να σημειωθεί πως η επιλογή του αρχικού (256×256) και τελικού μεγέθους (224×224) είναι βασισμένα στην προεπεξεργασία που έγινε στις εικόνες που χρησιμοποιήθηκαν για την εκπαίδευση των προεκπαιδευμένων μοντέλων που χρησιμοποιούνται. Στη παρούσα δουλειά, η περικοπή κυμαίνεται αναλόγως το μοντέλο cnn (backbone) που χρησιμοποιείται. Συγκεκριμένα, χρησιμοποιούνται τα μεγέθη 204×204, 190×190, 170×170 και 150×150. Η τυχαία περικοπή, όπως λέει και το όνομα, συμβαίνει κάθε φορά σε τυχαίο σημείο, με σκοπό να αποκόψει κάποιο κομμάτι πληροφορίας από το μοντέλο και να το ωθήσει να χρησιμοποιήσει άλλες χαρακτηριστικά για την εξαγωγή αποτελέσματος. Με αυτή τη μέθοδο καθορίζεται το μέγεθος της εικόνας για καθένα από τα μεγέθη που δοκιμάζονται.
- **Horizontal Flip**: Επίσης υλοποιήθηκε η μέθοδος horizontal flip με πιθανότητα 50 %. Αυτό σημαίνει πως κάθε εικόνα έχει 50 % πιθανότητα να αναποδογυριστεί πάνω στον οριζόντιο άξονά της.
- **Random Temporal Scaling**: Τέλος, υλοποιείται η μέθοδος της τυχαίας χρονικής κλιμάκωσης $\pm 20\%$), όπου το μέγεθος ενός βίντεο μεγαλώνει ή μικραίνει, σε αυτή τη περίπτωση κατά 20%, το οποίο γίνεται με την προσθήκη διπλότυπων καρέ ή την αφαίρεση ορισμένων, αντίστοιχα.

5.6.0.2 Προεπεξεργασία Greek Sign Language Dataset : Η διαδικασία που ακολουθείται για την προεπεξεργασία του GSL διαφέρει κυρίως στο πρώτο κομμάτι, που εκτελείται πριν την εκπαίδευση. Αρχικά, έπρεπε να γίνουν μερικές μικρές χειροκίνητες διορθώσεις στο αρχείο που περιέχει τις πληροφορίες της εκπαίδευσης που παρείχε το σύνολο, γιατί υπήρχαν ορισμένες λανθασμένες αντιστοιχίες βίντεο-μετάφρασης. Έπειτα, παρατηρήθηκε πως τα βίντεο κατέγραφαν μια μεγάλη περιοχή του χώρου, αφήνοντας αρκετό χώρο κενό γύρω από τον διερμηνέα. Αυτό σημαίνει πως από τα συνολικά εικονοστοιχεία κάθε εικόνας, ένα μεγάλο κομμάτι δεν περιλάμβανε καμία πληροφορία. Για αυτό το λόγο επιλέχθηκε να αποκοπεί η κενή περιοχή. Για αυτό το σκοπό, οι συγγραφείς παρείχαν ένα αρχείο με ακριβείς συντεταγμένες, που δημιουργούσαν ένα παραλληλόγραμμο ”ενδιαφέροντος” (bounding box) για κάθε βίντεο. Η υπόλοιπη μεθοδολογία παραμένει σταθερή όπως για το προηγούμενο σύνολο. Δημιουργούνται τα κατάλληλα αρχεία και στο δεύτερο στάδιο οι εικόνες πάλι τροποποιούνται με χρήση random crop, horizontal flip (50%) και random temporal scaling (±20%).

5.6.0.3 Τεχνικές που δοκιμάστηκαν : Κατά τη διάρκεια δοκιμών προκειμένου να επιτευχθούν τα καλύτερα δυνατά αποτελέσματα, δοκιμάστηκαν ορισμένες επιπλέον διαδικασίες για την προεπεξεργασία των συνόλων. Παρακάτω θα αναφερθούν αυτές οι διαδικασίες καθώς και οι λόγοι που δοκιμάστηκαν.

- **Αλλαγή αρχικού μεγέθους**: Στην αρχική αλλαγή μεγέθους των εικόνων, δοκιμάστηκε το μέγεθος 224×224 έναντι του 256×256. Αυτό ήταν μία προσπάθεια να μειωθεί το ποσοστό επιπλέον πληροφορίας που αποκοπτόταν λόγω της μικρότερης εισόδου που χρησιμοποιήθηκε.

Αναλυτικότερα, για τα μοντέλα SqueezeNet και ShuffleNetv2, όπου η είσοδος είναι 204×204, για αρχικό μέγεθος εικόνων:

- 256, αφαιρούνται 52 εικονοστοιχεία, δηλαδή το $20,3\%$ των αρχικών εικονοστοιχείων.
- 224, αφαιρούνται 20 εικονοστοιχεία, δηλαδή το $11,2\%$ των αρχικών εικονοστοιχείων.

Επίσης, η αποκοπή που χρησιμοποιείται αρχικά σε αυτό το μοντέλο, και σε πολλά άλλα, είναι 32 εικονοστοιχεία, και συγκεκριμένα για αρχικό μέγεθος 256×256, το $12,5\%$ του αρχικού μεγέθους. Συμπερασματικά, η αλλαγή μεγέθους των εικονών σε 224×224, οδηγεί σε χάσιμο λιγότερης πληροφορίας και είναι πιο κοντά στο 'ιδεατό' που χρησιμοποιείται από το αρχικό μοντέλο. Παρ'όλαυτά, τα εμπειρικά αποτελέσματα σε ορισμένες εκπαιδεύσεις που πραγματοποιήθηκαν, ήταν ελαφρώς χειρότερα και για αυτόν τον λόγο δε χρησιμοποιήθηκε.

- **Η χρήση ή όχι bounding box**: Ένας σημαντικός λόγος που υπήρχε αμφιβολία ως προς τη χρήση bounding box κατά την προεπεξεργασία των εικονών του GSL, ήταν πως και οι συγγραφείς στο αρχικό τους άρθρο δεν ανέφεραν τη χρήση του σε δοκιμές που πραγματοποίησαν. Επίσης, ο επιπλέον κενός χώρος που παρείχε, θα μπορούσε να εξισορροπήσει τα επιπλέον εικονοστοιχεία που περικόπτονται. Από την άλλη, μέσω της εκτεταμένης συρρίκνωσης, θα χανόταν ένα κομμάτι πληροφορίας. Όπως και στα άλλα παραδείγματα, δοκιμάστηκαν σε μερικές τυχαίες εκπαιδεύσεις και τα δύο σενάρια, και φάνηκε πως η χρήση bounding box οδηγούσε σε ελαφρώς καλύτερα αποτελέσματα, και καθώς η χρήση της δεν επιβάρυνε καθόλου το μοντέλο, αποφασίστηκε να χρησιμοποιηθεί.

- **Επαναδειγματοληψία**: Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο, το GSL είναι καταγεγραμμένο σε 30 fps, ενώ το phoenix14 σε 25. Καθώς ο χρονικός προσδιορισμός διαδραματίζει σημαντικό ρόλο στο μοντέλο, δοκιμάστηκε η επαναδειγματοληψία του GSL ώστε να προσομοιώνεται η καταγραφή σε 25 fps. Πάλι η μέθοδος δοκιμάστηκε σε ορισμένες εκπαιδεύσεις και επειδή τα αποτελέσματα δεν διέφεραν σημαντικά, αποφασίστηκε να διατηρηθεί η αρχική μορφή που ήταν και η πιο απλή.

5.7 Επιλογή δικτύου CNN για εξαγωγή χαρακτηριστικών

Η πρώτη πρόκληση που αντιμετωπίστηκε ήταν να τρέξει το μοντέλο και να ξεπεραστεί το πρόβλημα της έλλειψης αρκετής εικονικής μνήμης της κάρτας γραφικών. Να σημειωθεί πως τα πειράματα πραγματοποιήθηκαν σε κάρτα Nvidia GeForce 1660 Ti, 6GB, 16GB RAM σε λειτουργικό σύστημα Ubuntu 22.04 LTS. Για να καταπολεμηθεί το σφάλμα που δημιουργούνταν, εξετάστηκαν βελτιστοποιήσεις όπως η απαλοιφή των δεδομένων που είχαν φορτωθεί στην κάρτα μεταξύ των εποχών και διαγραφή περιττών μεταβλητών.

Αυτές οι αλλαγές δεν ήταν αρκετές οπότε στη συνέχεια δοκιμάστηκαν αλλαγές σε παραμέτρους. Η πρώτη αλλαγή ήταν η μείωση του μεγέθους παρτίδας (batch size) από δύο σε ένα. Αυτή η αλλαγή θα μεγάλωνε σημαντικά τον χρόνο εκπαίδευσης, αλλά θα προκαλούσε δραστική μείωση στην χρήση εικονικής μνήμης.

Η επόμενη αλλαγή που δοκιμάστηκε ήταν να τεθεί η μεταβλητή frame interval από ένα σε δύο, δηλαδή το μοντέλο να προσπερνάει ένα καρέ για κάθε καρέ που δέχεται στην είσοδο. Σε αυτή τη περίπτωση,

το μοντέλο μπόρεσε να εκτελεστεί αλλά η ασυνέχεια στις απώλειες και ο μικρός αριθμός εισόδων δεν επέτρεψαν στο μοντέλο να συγκλίνει.

Έπειτα από αυτές τις αλλαγές δοκιμάστηκε η μείωση των εισόδων στο μοντέλο, δηλαδή η μείωση του μεγέθους των εικόνων. Δοκιμάστηκαν οι τιμές 210, 200, 195 και 190. Από αυτές, μόνο η τελευταία μπορούσε να ολοκληρώσει σταθερά την εκπαίδευση, οπότε ήταν και η επιλογή για το δίκτυο Resnet18.

Η μείωση μεγέθους παρτίδας βελτίωσε τις συνθήκες αλλά και πάλι η κάρτα γραφικών δεν ήταν ικανή να ικανοποιήσει τις απαιτήσεις του μοντέλου. Επειδή μειώθηκε το μέγεθος παρτίδας και μάλιστα φτάνοντας στο κατώτατο όριο, επιλέχθηκε να δοκιμαστεί και η μείωση του ρυθμού εκμάθησης (learning rate - lr) για να καταπολεμηθεί ο θόρυβος που εισάγεται λόγω της αλλαγής. Επιπλέον, θεωρήθηκε πως ιδίως για το GSL το οποίο είναι εύκολο να κάνει overfit, όπως θα δούμε και παρακάτω, ένα μικρότερο lr θα επωφελούσε την εκμάθηση. Δε δοκιμάστηκαν πολλές τιμές λόγω χρόνου αλλά καθώς είναι δύσκολο να βρεθεί μία ιδανική τιμή, υποδιπλασιάζουμε την τιμή lr, από 0.0001 σε 0.00005. Παράλληλα αυξάνουμε λίγο τη τιμή weight_decay ώστε να ασκείται μεγαλύτερη ”ποινή” σε πολύ μεγάλα βάρη, πάλι για την αποφυγή του overfit. Με λίγες δοκιμές, φαίνεται πως αυτές οι τιμές βελτιώνουν λίγο τα αποτελέσματα, οπότε υιοθετούνται για τις εκπαιδεύσεις. Αντίθετα, στο σύνολο phoenix14, φαίνεται πως οι αρχικές τιμές είναι καλύτερες, οπότε για το phoenix14, διατηρούνται οι αρχικές. Σε κάθε περίπτωση, διαιρούμε το ρυθμό εκμάθησης με το 5, στις εποχές 20 και 35 αντίστοιχα, ώστε να μειωθούν οι μεγάλες διακυμάνσεις και να σταθεροποιηθεί το μοντέλο. Στην επόμενη υποενότητα φαίνονται αναλυτικά όλες οι μεταβλητές που χρησιμοποιήθηκαν.

Η μεγάλη μείωση που ήταν αναγκαίο να γίνει προκειμένου να τρέξει το μοντέλο, αποτέλεσε το έναυσμα για την διερεύνηση μοντέλων CNN που χρειαζόντουσαν λιγότερους υπολογιστικούς πόρους για να λειτουργήσουν, προκειμένου το πλεόνασμα πόρων να καταναλωθεί σε επιπλέον εισόδους. Στη βιβλιογραφία, συχνά τα συνελικτικά δίκτυα δύο διαστάσεων που χρησιμοποιούνται στο πρώτο κομμάτι του μοντέλου ως εξαγωγείς χαρακτηριστικών, αναφέρονται ως backbones. Για χάρη συντομίας αυτός ο όρος θα χρησιμοποιείται και στο παρόν κείμενο. Η διερεύνηση για εύρεση πιο ”ελαφριών” backbones με λιγότερες παραμέτρους, δικαιολογήθηκε περαιτέρω καθώς λόγω του μικρού, σχετικά, όγκου του συνόλου δεδομένων, έχουν τη δυνατότητα να λειτουργήσουν καλύτερα από ”βαριά” και να αποφύγουν το overfitting.

Εξετάστηκαν backbones από αυτά που προσφέρονται προεκπαιδευμένα στη βιβλιοθήκη PyTorch [51]. Τα κριτήρια με τα οποία έγινε η αρχική επιλογή ήταν ο αριθμός παραμέτρων, τα GFLOPS και το μέγεθος αρχείου να είναι σημαντικά μικρότερος από του resnet18. Έπειτα, ελέχθηκε η απόδοση των μοντέλων στο σύνολο δεδομένων ImageNet. Αυτά τα κριτήρια, τα πληρούσαν τα μοντέλα SqueezeNet1_1, ShuffleNetv2_x1_0, MobileNetv2, MobileNetv3 και MNASNET0_5. Με το MobileNetv3 πραγματοποιήθηκαν ορισμένες δοκιμές και ήταν το μόνο μοντέλο που επέτρεψε την εκτέλεση με κανονικό μέγεθος εισόδου (224×224), αλλά τα αποτελέσματα ήταν αρκετά αποθαρρυντικά. Έπειτα, το MobileNetv2 παρά το μικρό του μέγεθος, δεν κατάφερε να τρέξει. Από τα υπόλοιπα επιλέχθηκε το SqueezeNet1_1 για το ιδιαίτερα μικρό μέγεθος του και την ιδιαίτερη αρχιτεκτονική του. Το συγκεκριμένο, είναι μία ελαφρώς βελτιωμένη εκδοχή του SqueezeNet που αναλύθηκε σε προηγούμενο κεφάλαιο, καθώς προσφέρει την ίδια απόδοση αλλά με σημαντικά μειωμένους υπολογισμούς ($\times 2.4$), Επίσης επιλέχθηκε το ShuffleNetv2_1_0 λόγω της καλής του απόδοσης, σχεδόν όμοιας με του ResNet18, ενώ διατηρεί τους

υπολογισμούς του κάτω από αυτούς του SqueezeNet. Επιπροσθέτως, σύμφωνα με τους ισχυρισμούς των συγγραφέων του, είναι και πολύ γρήγορο, καθιστώντας το από τις καλύτερες επιλογές για λειτουργία σε πραγματικές συνθήκες.

Μοντέλο	**Αριθμός Παραμέτρων**	**GFLOPS**	**Μέγεθος Αρχείου (MB)**
Resnet18	11.689.512	1,81	44,70
ShuffleNetv2_1.0	2.278.604	0,14	8,80
SqueezeNet1_1	1.235.496	0,35	4,70

Πίνακας 5.1: Σύγκριση παραμέτρων, αναγκαίων υπολογισμών και μεγέθους αρχείου των επιλεγμένων συνελικτικών δικτύων

Μοντέλο	**top-5 Accuracy (%)**	**top-1 Accuracy (%)**
Resnet18	89,078	69,758
ShuffleNetv2_1.0	88,316	69,362
SqueezeNet1_1	80,624	58,178

Πίνακας 5.2: Σύγκριση απόδοσης των επιλεγμένων συνελικτικών δικτύων στο σύνολο ImageNet

5.8 Παράμετροι

Παρακάτω παρατίθενται αναλυτικά οι παράμετροι που χρησιμοποιήθηκαν για την εκτέλεση του μοντέλου. Οι πιο αξιοσημείωτες για το μοντέλο είναι οι:

- num_epoch: Ο αριθμός εποχών που εκπαιδεύεται το μοντέλο. Στη προκειμένη περίπτωση το θέτουμε ίσο με 40, αλλά για ορισμένες εκπαιδεύσεις σταματήθηκε πρόωρα, ακολουθώντας την λογική πως αν μετά την 25η εποχή, δεν γίνει κάποια βελτίωση για πέντε συνεχόμενες εποχές, τότε η εκπαίδευση σταματάει. Αυτό υλοποιήθηκε μόνο στο σύνολο δεδομένων Phoenix14, καθώς είχε μικρές διακυμάνσεις προς το τέλος της εκπαίδευσης και δημιουργούταν ένα πλάτωμα, ενώ το GSL, λόγω αρκετά μεγαλύτερων διακυμάνσεων ήταν πιο απρόβλεπτο, αν και προς το τέλος παρουσίαζε συχνά overfitting.
- batch_size: Το μέγεθος παρτίδας ορίζεται ως ένα, οπότε κάθε φράση επεξεργάζεται μεμωνομένα.
- loss_ weights: Εδώ μπορεί να οριστεί η χρήση ή όχι των επιπλέον συναρτήσεων απώλειας του μοντέλου. Το SeqCTC είναι η κλασική συνάρτηση απώλειας CTC, ενώ η ConvCTC και η Dist ενεργοποιούν τις επιπλέον συναρτήσεις VA και VE.
- optimizer: Αυτή καθορίζει τον αλγόριθμο βελτιστοποίησης των βαρών του CNN. Εδώ χρησιμοποιείται ο αλγόριθμος Adam.
- base_lr: Είναι ο ρυθμός εκμάθησης του μοντέλου, ή βήμα. Για εκπαίδευση στο σύνολο Phoenix14, τίθεται ίσο με 0,0001, ενώ για εκπαίδευση στο σύνολο GSL, μειώνεται και τίθεται ίσο με 0,00005.
- step: Οι εποχές στις οποίες ο ρυθμός εκμάθησης διαιρείται με το πέντε.
- weight_decay: Ορίζει την ”ποινή” που δέχεται το μοντέλο όταν παράγει υπερβολικά μεγάλα βάρη. Για εκπαίδευση στο σύνολο Phoenix14, τίθεται ίσο με 0,0001, ενώ για εκπαίδευση στο σύνολο GSL, διπλασιάζεται και τίθεται ίσο με 0,0002.

- inputs: Το μέγεθος των εικόνων που δέχεται ως εισόδους το μοντέλο. Κάθε εικόνα έχει μέγεθος $(inputs \times inputs \times 3)$.

Υπάρχουν δύο επιπλέον σημαντικές μεταβλητές οι οποίες δεν βρίσκονται σε αυτό το σημείο του κώδικα. Αυτές είναι το μέγεθος εισόδου που δέχεται το δίκτυο BiLSTM. Αυτή ονομάζεται *hidden_size* και τίθεται ίση με 1024, το οποίο είναι ίσο με 2×512, λόγω των δύο κατευθύνσεων του δικτύου. Συγκεκριμένα, οι πρώτες 512 είσοδοι, επεξεργάζονται από την πρώτη προς την τελευταία. Παράλληλα, οι άλλες 512 επεξεργάζονται από τη τελευταία προς τη πρώτη. Στο τέλος, οι δύο έξοδοι ενώνονται για να δημιουργήσουν το τελικό αποτέλεσμα. Η άλλη μεταβλητή είναι το *dropout* που εφαρμόζεται στο BiLSTM δίκτυο, το οποίο παίρνει τη τιμή 0,3.

```
feeder: dataset.dataloader_video.BaseFeeder
phase: train
dataset: phoenix14 #continuous_gsl
# dataset: phoenix14-si5
num_epoch: 40
work_dir: ./work_dir/baseline_res18/
batch_size: 1
random_seed: 0
test_batch_size: 1
num_worker: 10
device: 0,1
log_interval: 100
eval_interval: 1
save_interval: 5
# python in default
evaluate_tool: sclite
loss_weights:
  SeqCTC: 1.0
  # VAC
  ConvCTC: 1.0
  Dist: 10.0
#load_weights: ''

optimizer_args:
  optimizer: Adam
  base_lr: 0.0001 #0.00005
  step: [ 20, 35]
  learning_ratio: 1
  weight_decay: 0.0001 #0.0002
  start_epoch: 0
  nesterov: False
```

```
feeder_args:
  mode: 'train'
  datatype: 'video'
  frame_interval: 1
  num_gloss: -1
  drop_ratio: 1.0
  input_size: 170 #224 204 190

model: slr_network.SLRModel
decode_mode: beam
model_args:
  num_classes: 1296 #314
  c2d_type: resnet18       #resnet18  squeezenet1_1 shufflenet_v2_x1_0
  conv_type: 2
  use_bn: 1
  # SMKD
  share_classifier: False
  weight_norm: False
```

5.9 Μετρικές

Στη παρούσα εργασία, οι μετρικές που αξιοποιούνται, αποσκοπούν στην κάλυψη τριών βασικών κριτηρίων για μοντέλα αναγνώρισης νοηματικής. Αυτά είναι η απόδοση, η καταλληλότητα για χρήση σε πραγματικές χρονικές συνθήκες και αν μπορούν να αξιοποιηθούν σε εφαρμογές του πραγματικού κόσμου, από την σκοπιά των υλικών απαιτήσεων που απαιτούνται. Για την κάλυψη της απόδοσης, χρησιμοποιείται η μετρική Word Error Rate (WER). Αντίστοιχα, για την κάλυψη των χρονικών αναγκών και της αξιοποίησης σε εφαρμογές, αξιοποιούνται τα FLOP και η ταχύτητα επεξεργασίας φράσης. Ακολουθεί μια πλήρης ανάλυση των μετρικών.

5.9.0.1 Word Error Rate : Το Word Error Rate (WER) είναι από τις πιο κοινές μετρικές που χρησιμοποιείται για την αξιολόγηση των συστημάτων αναγνώρισης ομιλίας και η πιο διαδεδομένη για την συνεχή αναγνώριση νοηματικής. Αυτό που κάνει, είναι να μετράει πόσες πράξεις χρειάζονται ώστε να μετατραπεί η προβλεπόμενη σειρά λέξεων στην αρχική πρόταση-στόχο. Ο τύπος για τον υπολογισμό του WER είναι: $WER = \frac{S+D+I}{N}$ όπου:

- **S**: Ο αριθμός των αντικαταστάσεων (πόσες λέξεις της πρότασης-εξόδου είναι διαφορετικές από την πρόταση-στόχο)
- **D**: Ο αριθμός διαγραφών (πόσες λέξεις του της πρότασης-στόχου δεν υπάρχουν στην πρόταση-έξοδο)
- **I**: Ο αριθμός εισαγωγών (πόσες λέξεις στην πρόταση-έξοδο δεν υπάρχουν στην πρόταση-στόχο)
- **N**: Ο συνολικός αριθμός λέξεων της πρότασης-στόχο

5.9.0.2 Floating Point Operation (FLOP) : Αποτελεί μία από τις πιο συνηθισμένες μονάδες μέτρησης υπολογιστικού κόστους, ιδίως στον τομέα των νευρωνικών δικτύων, και περιλαμβάνει τη μέτρηση των πράξεων πρόσθεσης, πολλαπλασιασμού, διαίρεσης. Για παράδειγμα η πράξη $z * x + y$ αποτελεί δύο FLOP. Η πιο συχνά χρησιμοποιούμενη μονάδα μέτρησης των FLOP είναι τα Gigaflop (GFLOP), όπου 1 GFLOP, αντιστοιχεί σε 10^9 flop. Αναφέρεται επίσης πως μπορεί να συναντηθεί και στη μορφή FLOPs, το οποίο όμως σημαίνει Floating Point Operations per second. Η θέση του ως μία καλή μονάδα μέτρησης είναι αδιαμφισβήτητη, αλλά όπως αναφέρθηκε παραπάνω, οι συγγραφείς στο [45] πιστεύουν πως δεν αντικατοπτρίζει πλήρως τις δυνατότητες ενός μοντέλου. Ένα παράδειγμα, είναι πως μοντέλα με χαμηλότερα FLOP, μπορεί είναι πιο αργά από μοντέλα που απαιτούν περισσότερα. Για αυτό το λόγο, χρησιμοποιείται και η παρακάτω μετρική για πληρότητα.

5.9.0.3 Ταχύτητα επεξεργασίας παρτίδας : Επειδή ένας από τους στόχους είναι να ελεγχθεί η καταλληλότητα των μοντέλων για χρήση σε πραγματικό χρόνο και πραγματικό περιβάλλον, ο χρόνος που απαιτείται για να επεξεργαστούν οι φράσεις, θεωρείται ένα απαραίτητο μέγεθος. Για αυτό υιοθετείται η μετρική παρτίδα/δευτερόλεπτο, το οποίο για το συγκεκριμένο μοντέλο είναι ίσο με τη φράση/δευτερόλεπτο (it/s) (το μέγεθος παρτίδας έχει τεθεί ίσο με τη μονάδα). Αυτό μπορεί να εξηγηθεί ως, πόσες φράσεις επεξεργάζεται το μοντέλο σε ένα δευτερόλεπτο, όπου κάθε φράση είναι και ένα βίντεο.

Κεφάλαιο 6ο: Αποτελέσματα

Σε αυτό το κεφάλαιο θα παρουσιαστούν και θα περιγραφούν τα αποτελέσματα που προέκυψαν από τις πολλαπλές εκπαιδεύσεις στα σύνολα δεδομένων που πραγματοποιήθηκαν και θα γίνει. Ο σχολιασμός των αποτελεσμάτων και τα συμπεράσματα που εξάγονται, θα περιγραφούν στο επόμενο κεφάλαιο. Για τη διευκόλυνση της κατανόησης των αποτελεσμάτων, χωρίζονται ως προς τις τρεις βασικές κατηγορίες που έγινε η προσπάθεια να μετρηθούν. Η πρώτη είναι τα αποτελέσματα που αφορούν την απόδοση των δοκιμασμένων μοντέλων, ενώ η δεύτερη και τρίτη προσφέρει αποτελέσματα σχετικά με τις μετρικές που επιλέχθηκαν για τη μέτρηση των υπολογιστικών απαιτήσεων και της ταχύτητας εκτέλεσης. Τέλος, υπάρχει ένα επιπλέον τμήμα στο οποίο παρουσιάζονται κάποια δεδομένα για τις εκπαιδεύσεις, καθώς και κάποιες ενδεικτικές έξοδοι του μοντέλου.

6.1 Αποτελέσματα σχετικά με την απόδοση

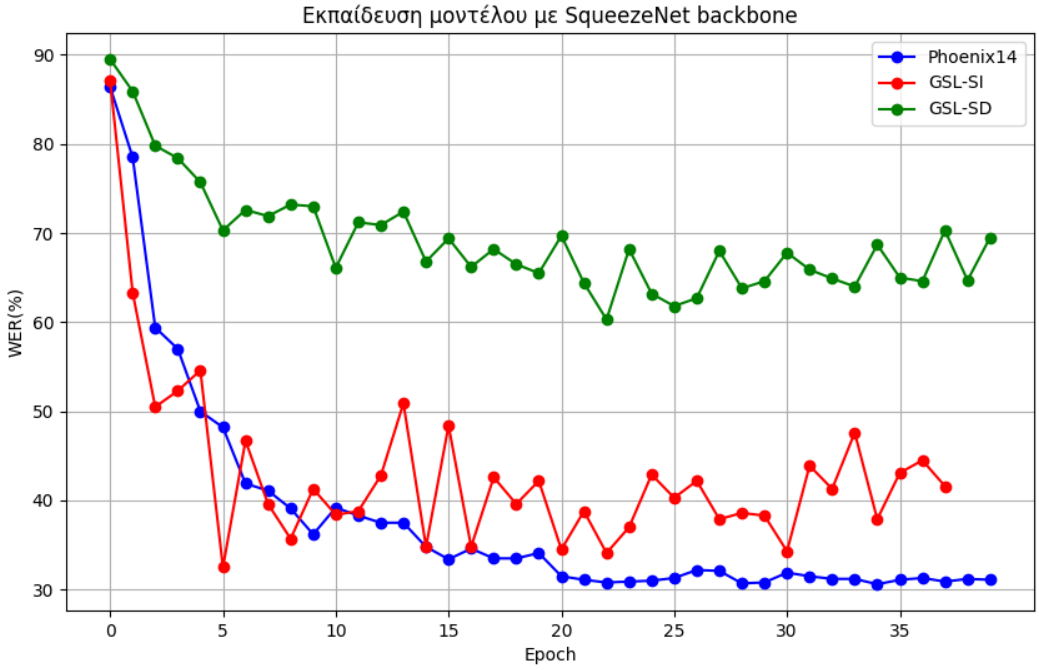
Στον πίνακα 6.2 φαίνεται μια αναλυτική σύγκριση των ποσοστών WER των μοντέλων, τόσο κατά τη διάρκεια της αξιολόγησης όσο και του ελέγχου, στα σύνολα δεδομένων GSL-SI, GSL-SD και Phoenix14. Οι αριθμοί που υπάρχουν στη στήλη inputs, είναι το ύψος και το πλάτος των εικόνων που δέχεται το μοντέλο ως εισόδους, σε εικονοστοιχεία. πχ. Για είσοδο 224, το μοντέλο δέχεται εικόνες με σχήμα $224 \times 224 \times 3$, όπου το 3 είναι αριθμός καναλιών των RGB εικονών. Τα μοντέλα που περιλαμβάνει, είναι αρχικά το VAC_CSLR με τα τρία διαφορετικά backbones που ελέγχθηκε, με τις μέγιστες τιμές εισόδων που υπήρχε η δυνατότητα να τρέξει. Έπειτα, περιλαμβάνει το αρχικό μοντέλο VAC_original (με ResnNet18 και εισόδους $224 \times 224 \times 3$), και δύο ακόμα από τα μοντέλα με την καλύτερη απόδοση στην αναγνώριση νοηματικής. Τέλος, περιλαμβάνει τα μοντέλα τα οποία αναφέρονται στο [26], τα οποία αποτέλεσαν και το αρχικό σημείο αναφοράς για το σύνολο δεδομένων GSL.

Πίνακας 6.1: Σύγκριση Word Error Rate (WER) για διαφορετικές εισόδους στο σύνολο Phoenix14

Backbone	**Inputs**	**Test WER (%)**	**Dev WER (%)**
ResNet18	224	22,30	21,20
ResNet18	204	-	-
SqueezeNet1_1	204	**31,40**	**30,60**
ShuffleNet_v2_x1_0	204	35,80	37,50
ResNet18	190	**31,90**	**31,10**
SqueezeNet1_1	190	32,40	31,70
ShuffleNet_v2_x1_0	190	34,70	35,10
ResNet18	170	**30,10**	**29,80**
SqueezeNet1_1	170	33,10	32,70
ShuffleNet_v2_x1_0	170	34,40	33,80
ResNet18	150	**30,10**	**31,10**
SqueezeNet1_1	150	32,90	33,50
ShuffleNet_v2_x1_0	150	34,70	35,80

Από αυτόν τον πίνακα μπορούμε να παρατηρήσουμε πως οι παραλλαγές που δοκιμάστηκαν παρουσιάζουν αποτελέσματα με μικρή, σχετικά, απόκλιση όταν εκπαιδεύονται στα σύνολα GSL-SI και Phoenix14. Αντίθετα, όταν η εκπαίδευση γίνεται πάνω στο σύνολο GSL-SD, τα αποτελέσματα είναι πολύ χειρότε-

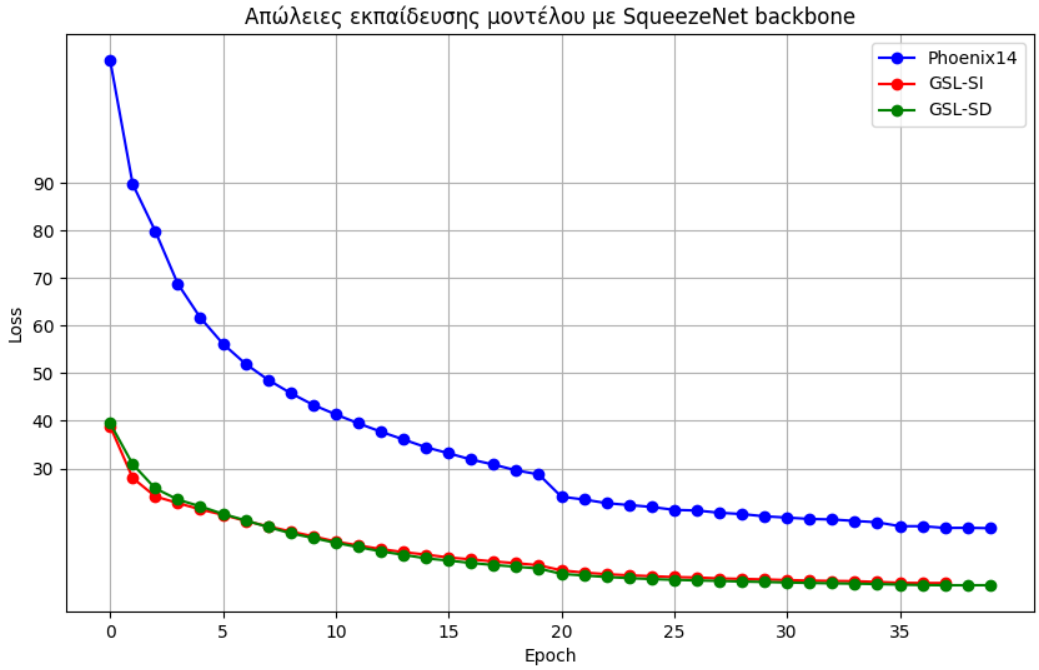
ρα, και πάλι σε ορισμένα δίκτυα, διπλασιάζουν το ποσοστό WER από άλλα σύνολα. Για παράδειγμα, το μοντέλο με δίκτυο SqueezeNet πετυχαίνει 32.0% και 31.4% WER στο σύνολο ελέγχου των GSL-SI και GSL-SD, ενώ στο αντίστοιχο σύνολο του GSL-SD πετυχαίνει μόλις 67.0%. Επίσης, από τις παραλλαγές που δοκιμάστηκαν με τις μέγιστες δυνατές εισόδους, το SqueezeNet παρουσιάζει τα καλύτερα αποτελέσματα σε όλα τα σύνολα δεδομένων. Δεύτερο είναι το ResNet18, με εξαίρεση το σύνολο GSL-SD, όπου το ShuffleNetv2 το προσπερνάει.



Σχήμα 6.1: Η εκπαίδευση του μοντέλου με SqueezeNet backbone στα τρία σύνολα δεδομένων

Έπειτα, στον πίνακα 6.1 παρατίθενται τα αποτελέσματα που εξάχθηκαν από τις εκπαιδεύσεις των παραλλαγών του μοντέλου με διαφορετικά μεγέθη εισόδων. Από αυτόν τον πίνακα, μπορούμε να παρατηρήσουμε πως το ResNet18, παρά την μεγάλη αρχική πτώση σε απόδοση μεταξύ των μεγεθών εισόδων 224 και 190, έχει συστηματικά τη καλύτερη απόδοση σε όσες κατηγορίες εισόδων συμμετέχει. Να διευκρινιστεί, πως στη προηγούμενη σύγκριση του πίνακα 6.2, το SqueezeNet ήταν καλύτερο, επειδή συγκρινόταν με το ResNet18 με εισόδους 190, ενώ εκείνο είχε 204. Οπότε σε σύγκριση για ίδιες εισόδους,το SqueezeNet είναι δεύτερο, για πολύ μικρή διαφορά στην αρχή (0.6%) που διευρύνεται όσο μικραίνουν οι είσοδοι. Αυτό οδηγεί σε μια ακόμα πολύ ενδιαφέρουσα παρατήρηση. Το ResNet18, μετά την αρχική πτώση απόδοσης, βελτιώνει την απόδοση του όταν οι είσοδοι μικραίνουν περαιτέρω. Το ίδιο συμβαίνει και στο ShuffleNetv2, το οποίο από την μέγιστη του είσοδο, όσο μειώνεται το μέγεθος των εισόδων, βελτιώνεται η απόδοση του, ενώ αντίθετα, η απόδοση του SqueezeNet μειώνεται.

Στη συνέχεια, στο σχήμα 6.1, παρουσιάζεται ένα γράφημα της εκπαίδευσης του μοντέλου με backbone το SqueezeNet. Κάθε γραμμή υποδεικνύει το WER του μοντέλου, κατά τον έλεγχό του στο σύνολο αξιολόγησης, στην εκάστοτε εποχή για το αντίστοιχο χρώμα, με μπλε τη γραμμή για την εκπαίδευση στα δεδομένα του Phoenix14, πράσινη για το GSL-SD και κόκκινη για το GSL-SI. Με παρόμοιο τρόπο φαίνονται οι απώλειες του μοντέλου κατά την εκπαίδευση στο επόμενο σχήμα 6.2, με τη διαφορά πως



Σχήμα 6.2: Οι απώλειες κατά την εκπαίδευση του μοντέλου με SqueezeNet backbone στα τρία σύνολα δεδομένων

οι απώλειες μετριούνται με το σύνολο εκπαίδευσης. Η πρώτη παρατήρηση που γίνεται είναι πως η εκπαίδευση στα δεδομένα του Phoenix14 είναι πολύ πιο ομαλή, με μικρές διακυμάνσεις και πολύ μικρές αυξομειώσεις μετά την τριακοστή εποχή. Επίσης παρατηρείται πως η εκπαίδευση στο GSL-SD φέρνει πολύ υψηλά ποσοστά λάθους, το οποίο λειτουργεί ως ένδειξη πως υπάρχει κάποιο πρόβλημα. Τέλος, η εκπαίδευση στο GSL-SI παράγει ασταθή αποτελέσματα, με το μοντέλο να εμφανίζει μεγάλες διακυμάνσεις, ακόμη και όταν χρησιμοποιείται μειωμένος ρυθμός εκμάθησης. Επιπλέον, παρατηρείται μόνο μικρή βελτίωση σε αυτόν τον τομέα προς το τέλος της εκπαίδευσης, παρά τη περαιτέρω μείωση του ρυθμού εκμάθησης. Παρ' όλα αυτά, σε ορισμένες εποχές, έχει επιτευχθεί σχετικά καλή απόδοση. Αυτές οι παρατηρήσεις, σε συνδυασμό με το σχήμα 6.2 όπου φαίνεται πως οι απώλειες φτάνουν πολύ γρήγορα, πολύ χαμηλές τιμές στα σύνολα GSL-SI και GSL-SD, είναι σημάδια overfitting. Το μοντέλο δηλαδή έχει μάθει "απέξω" πολλές από τις φράσεις και τις συνθήκες και δεν αλλάζει τα βάρη του, ενώ δεν έχει εξάγει ουσιαστικά χαρακτηριστικά που να το βοηθούν να κάνει προβλέψεις πάνω σε βίντεο που δεν έχει δει.

Πίνακας 6.2: Σύγκριση Word Error Rate (WER)

Backbone	**Inputs**	**GSL SI - bbox**		**GSL SD - bbox**		**Phoenix SD**	
		Dev WER (%)	**Test WER (%)**	**Dev WER (%)**	**Test WER (%)**	**Dev WER (%)**	**Test WER (%)**
TwoStream-SLR	224	-	-	-	-	18,40	18,80
CorrNet+	224	-	-	-	-	18,00	**18,20**
VAC_original	224	-	-	-	-	21,50	22,10
SubUNets	224	24,64	24,03	52,79	54,31	30,51	30,62
GoogLeNet+TConvs	224	08,08	**07,95**	43,54	**48,46**	32,18	31,37
3D-ResNet+BLSTM	224	33,61	33,07	61,94	68,54	38,81	37,79
I3D+BLSTM	224	08,78	08,62	51,74	53,48	32,88	31,92
SqueezeNet1_1	204	32,50	**32,00**	60,30	**67,00**	30,60	**31,40**
ShuffleNet_v2_x1_0	204	38,30	37,30	58,60	68,50	37,50	35,80
ResNet18	190	37,30	34,70	59,90	70,00	31,10	31,90

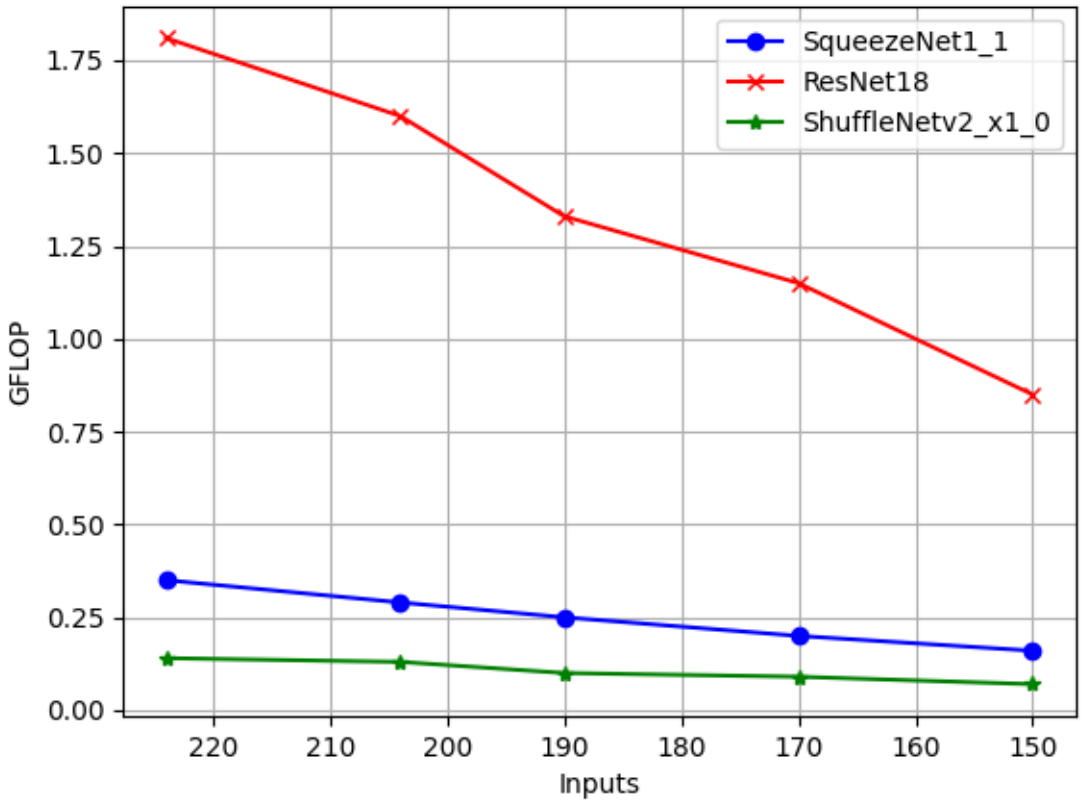
6.2 Αποτελέσματα σχετικά με το υπολογιστικό κόστος

Στον πίνακα 6.3 παρουσιάζονται τα GFLOP του κάθε backbone. Αυτή η μέτρηση δεν αντικατοπτρίζει τα GFLOP που χρειάζονται για όλο το το μοντέλο, αλλά μόνο για τα συγκεκριμένα δίκτυα έτσι όπως προσφέρονται από τη βιβλιοθήκη PyTorch, προεκπαιδευμένα στο σύνολο δεδομένων ImageNet. Αυτό έγινε καθώς ήταν δύσκολη η απευθείας μέτρηση ολόκληρου του μοντέλου, λόγω της πολυπλοκότητας και του τρόπου δομής του. Για τη μέτρηση χρησιμοποιήθηκε η βιβλιοθήκη *calflops*. Όμως, για σκοπούς σύγκρισης, παραμένει μία σημαντική μετρική, καθώς τα συνολικά GFLOP του μοντέλου μπορούν να θεωρηθούν ως $GFLOP_{total} = GFLOP_{backbone} + x$, όπου το x είναι περίπου σταθερό. Επιπλέον, επειδή ο αριθμός των GFLOP διαφέρει εάν η μέτρηση συμπεριλάβει μόνο τις πράξεις που γίνονται κατά το πέρασμα προς τα εμπρός ($GFLOP_{fwd}$) ή το πέρασμα προς το εμπρός και το πέρασμα προς τα πίσω. ($GFLOP_{fwd+backwd}$), συμπεριλήφθηκαν και τα δύο για σκοπούς πληρότητας. Από αυτά, τα $GFLOP_{fwd+backwd}$ αφορούν τη διαδικασία της εκπαίδευσης, ενώ τα $GFLOP_{fwd}$ αφορούν τη διαδικασία ελέγχου, δηλαδή μόνο την πρόβλεψη. Στον πίνακα τα αποτελέσματα παρέχονται με την εξής μορφή : $GFLOP_{fwd}/GFLOP_{fwd+backwd}$.

Επίσης, στο σχήμα 6.3, παρουσιάζεται ένα διάγραμμα που απεικονίζει τη μείωση των GFLOP των συνελικτικών δικτύων ανάλογα με το μέγεθος των εισόδων. Το διάγραμμα δημιουργήθηκε προκειμένου να γίνουν προφανής οι διαφορές στο μέγεθος των απαραίτητων GFLOP και ταυτόχρονα να τονίσει τη διαφορά που κάνουν οι είσοδοι στις υπολογιστικές απαιτήσεις του δικτύου. Είναι εμφανές, πως το ResNet18, το οποίο απαιτεί τα περισσότερα GFLOP, κάνει αριθμητικά την μεγαλύτερη πτώση όσο μειώνεται το μέγεθος και συγκεκριμένα μειώνεται κατά περίπου 53% μεταξύ της μεγαλύτερης και μικρότερης εισόδου. Η διαφορά μεταξύ μεγαλύτερης και μικρότερης εισόδου του SqueezeNet και του ShuffleNetv2 είναι σαφώς μικρότερη από του ResNet18, αλλά και αυτή αντιπροσωπεύει μια πτώση $50 - 55\%$ από τις αρχικές τους τιμές.

Input Size	ResNet18 GFLOP (f/b)	SqueezeNet1_1 GFLOP (f/b)	ShuffleNetV2_x1_0 GFLOP (f/b)
224	1,81/5,44	0,35/1,04	0,14/0,43
204	1,60/4,80	0,29/0,89	0,13/0,39
190	1,33/3,99	0,25/0,75	0,10/0,32
170	1,15/3,45	0,20/0,62	0,09/0,28
150	0,85/2,62	0,16/0,49	0,07/0,22

Πίνακας 6.3: Σύγκιριση GFLOP για κάθε ένα από τα backbones που χρησιμοποιήθηκαν με διαφορετικό αριθμό εισόδων. Ο πρώτος αριθμός υποδεικνύει τα GFLOP που απαιτούνται στο πέρασμα προς τα εμπρός, ενώ ο δεύτερος αυτά που απαιτούνται στο πέρασμα προς τα πίσω.



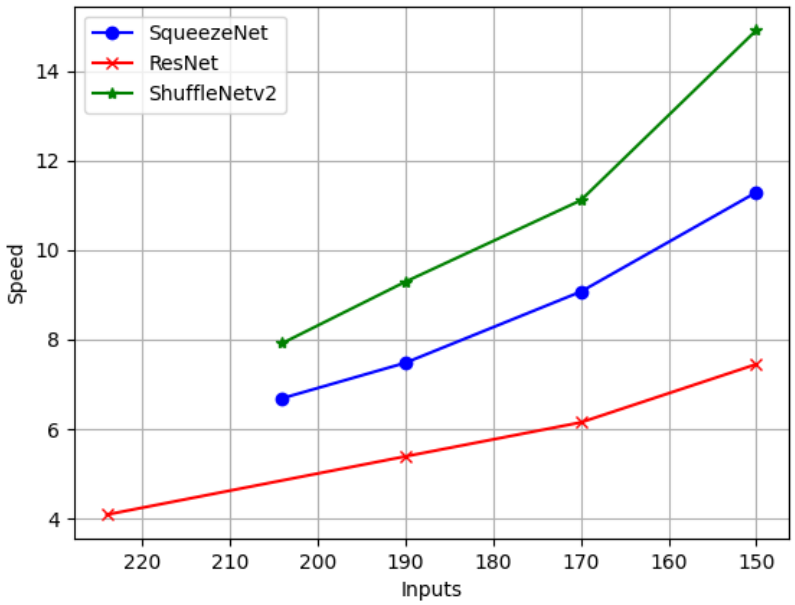
Σχήμα 6.3: Σύγκριση GFLOP των ResNet18, SqueezeNet1_1, Shufflenetv2_x1_0 για διαφορετικές εισόδους

6.3 Αποτελέσματα σχετικά με την ταχύτητα

Στους πίνακες 6.5, 6.6 και 6.7 παρουσιάζονται τα δεδομένα σχετικά με την ταχύτητα επεξεργασίας παρτίδας του μοντέλου με όλα τα δίκτυα backbone και όλες τις τιμές εισόδων που δοκιμάστηκαν, στα τρία σύνολα ή υποσύνολα δεδομένων. Η πρώτη σειρά αφορά το αρχικό μοντέλο, για το οποίο όμως δεν υπάρχουν δεδομένα ταχύτητας της εκπαίδευσης, καθώς όπως αναφέρθηκε σε προηγούμενη ενότητα, δεν μπόρεσε να εκτελεστεί, και οι συγγραφείς δεν παρέχουν τέτοια δεδομένα. Συγκεκριμένα, σε κάθε στήλη υπάρχει η ταχύτητα, μετρημένη σε παρτίδες ανά δευτερόλεπτο(it/s) , που αναλογεί σε καθένα από τα κομμάτια της διαδικασίας εκπαίδευσης. Επίσης, καταγράφεται το μέγεθος των βαρών του μοντέλου σε MegaByte (MB). Αυτό το αρχείο περιλαμβάνει πληροφορίες όπως η εποχή στην οποία βρισκόταν, την κατάσταση του optimizer, την τιμή που είχε δοθεί στη γεννήτρια τυχαίων αριθμών, ώστε να επιτρέπει την επαναληψιμότητα των πειραμάτων, και φυσικά τα βάρη του μοντέλου. Το μέγεθος αυτό είναι ενδεικτικό της γενικής πολυπλοκότητας του μοντέλου και παράλληλα οι πιο χαμηλές τιμές, καθιστούν πιο εύκολη την αξιοποίηση του.

Σύνολο Δεδομένων	Σύνολο Εκπαίδευσης	Σύνολο Αξιολόγησης	Σύνολο Ελέγχου
GSL SD	8189	1063	1043
GSL SI	8822	588	881
Phoenix14 MultiSigner	5671	540	629

Πίνακας 6.4: Αριθμός προτάσεων που περιλαμβάνει κάθε μέρος της διαδικασίας εκπαίδευσης (train, dev, test), για κάθε σύνολο δεδομένων.



Σχήμα 6.4: Οι ταχύτητες(it/s) των παραλλαγών στον σύνολο ελέγχου σε σχέση με τα μεγέθη εισόδων

Μία πρώτη παρατήρηση που γίνεται, είναι πως η εκπαίδευση έχει συστηματικά πολύ χαμηλότερη ταχύτητα, ανεξαρτήτως δικτύου CNN, από όλα τα υπόλοιπα κομμάτια. Αυτό συμβαίνει για δύο λόγους:

1. Κατά τη διάρκεια της εκπαίδευσης, το μοντέλο πραγματοποιεί περάσματα προς τα εμπρός και προς

τα πίσω (forward and backward passes). Το τελευταίο είναι και το πιο υπολογιστικά ακριβό γιατί υπολογίζει και ανανεώνει τα βάρη όλου του μοντέλου. Αντίθετα, στον έλεγχο και στην αξιολόγηση, πραγματοποιείται μόνο το πέρασμα μπρος στα εμπρός, κατά το οποίο χρησιμοποιούνται τα βάρη για να γίνουν οι προβλέψεις.

2. Όπως αναφέρθηκε πιο πάνω, κατά τη φόρτωση δεδομένων για την εκπαίδευση, πραγματοποιείται το δεύτερο μέρος της προεπεξεργασίας τους, το οποίο περιλαμβάνει διάφορες τροποποιήσεις των εικόνων. Αυτή η διαδικασία απαιτεί ένα επιπλέον υπολογιστικό κόστος, το οποίο μεταφράζεται και στη συνολική ταχύτητα επεξεργασίας. Αντίθετα, στα άλλα κομμάτια, οι εικόνες φορτώνονται αυτούσιες, με τη μοναδική επεξεργασία να είναι η κατάλληλη κεντρική περικοπή, εξαλείφοντας αυτό το κόστος.

Έπειτα, παρατηρούμε πως ένα μοντέλο με ίδιες εισόδους, μπορεί να έχει διαφορετικές ταχύτητες, αναλόγως με το σύνολο δεδομένων στο οποίο εξετάζεται. Ο λόγος για αυτό είναι πως κάθε σύνολο έχει διαφορετικό σύνολο από φράσεις για κάθε κομμάτι, με αποτέλεσμα να παρατηρούνται αποκλίσεις για λόγους που αφορούν τον τρόπο με τον οποίο φορτώνονται και αξιοποιούνται τα δεδομένα από την κάρτα γραφικών. Στον πίνακα 6.4 υπάρχει ο αναλυτικός διαχωρισμός του αριθμού προτάσεων που εξετάζονται σε κάθε κομμάτι, από κάθε σύνολο δεδομένων.

Πίνακας 6.5: Αποτελέσματα ταχύτητας για GSL SI

Backbone	**Inputs**	**Test Speed (it/s)**	**Dev Speed (it/s)**	**Train Speed (it/s)**	**Weights size (MB)**
squeezenet1_1	204	09,47	09,39	04,12	298,10
shufflenet_v2_x1_0	204	10,88	11,31	05,34	299,80
resnet18	190	07,64	07,63	02,86	387,40

Πίνακας 6.6: Αποτελέσματα ταχύτητας για GSL SD

Backbone	**Inputs**	**Test Speed (it/s)**	**Dev Speed (it/s)**	**Train Speed (it/s)**	**Weights size (MB)**
squeezenet1_1	204	08,93	07,48	04,28	298,20
shufflenet_v2_x1_0	204	10,91	09,19	05,52	299,90
resnet18	190	07,00	06,15	02,94	387,50

Από τα δίκτυα CNN που εξετάστηκαν, το ShuffleNetv2, είναι συστηματικά το πιο γρήγορο, παρά το γεγονός πως έχει περισσότερες παραμέτρους από το SqueezeNet, το οποίο είναι το δεύτερο πιο γρήγορο, ενώ το ResNet18 είναι τελευταίο. Επίσης παρατηρούμε πως η διαφορά μεταξύ πρώτου και δεύτερου, διευρύνεται καθώς οι είσοδοι μειώνονται.

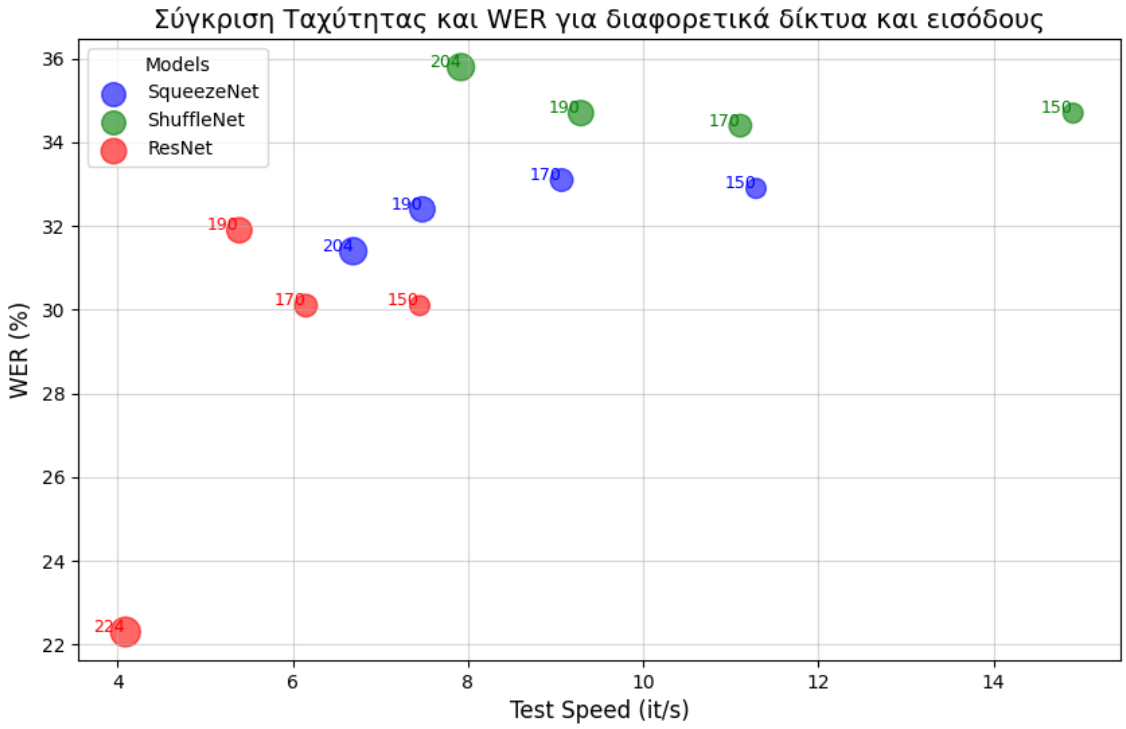
Κάτι ακόμα που παρατηρούμε είναι πως το μέγεθος του αρχείου διαφέρει από μοντέλο σε μοντέλο, αλλά παραμένει σταθερό ανεξάρτητα από το μέγεθος των εισόδων. Αυτό οφείλεται στις παραμέτρους, ο αριθμός των οποίων δεν επηρεάζεται από το μέγεθος των εισόδων.

Παρακάτω, στο σχήμα 6.6, εμφανίζονται συγκεντρωτικά όλα τα αποτελέσματα του μοντέλου στο σύνολο Phoenix14, με όλες τις παραλλαγές που πραγματοποιήθηκαν, σε σύγκριση με τα αποτελέσματα της αρχικής μορφής του. Για παράδειγμα, για το μοντέλο με δίκτυο ResNet18 και 170 εισόδους, βλέπουμε πως στο υποσύνολο ελέγχου, έχει χειρότερο ποσοστό WER κατά 7.8%, αλλά καλύτερη ταχύτητα κατά 2.06(it/s), σχεδόν ίδιο μέγεθος αρχείου, ενώ οι απαιτήσεις πέφτουν κατά 0.66 GFLOP. Σκοπός αυτού του

Πίνακας 6.7: Αποτελέσματα ταχύτητας για Phoenix14 Multisigner

Backbone	Inputs	Test Speed (it/s)	Dev Speed (it/s)	Train Speed (it/s)	Weights size (MB)
resnet18	224	04,09	04,23	-	411,80
squeezenet1_1	204	06,69	06,96	02,97	322,40
shufflenet_v2_x1_0	204	07,92	09,16	03,85	324,20
resnet18	190	05,39	05,55	02,02	411,70
squeezenet1_1	190	07,48	07,98	03,36	411,70
shufflenet_v2_x1_0	190	09,29	10,13	04,52	324,20
resnet18	170	06,15	06,44	02,35	411,70
squeezenet1_1	170	09,07	09,84	03,78	411,70
shufflenet_v2_x1_0	170	11,11	12,14	05,06	324,20
resnet18	150	07,45	07,97	02,91	411,70
squeezenet1_1	150	11,29	12,23	04,65	411,70
shufflenet_v2_x1_0	150	14,91	15,05	06,20	324,20

σχήματος είναι να παρέχει με πλήρως κατανοητό τρόπο τα θετικά και τα αρνητικά που παρέχει ο κάθε συνδυασμός δικτύου και αριθμού εισόδων από αυτά που δοκιμάστηκαν.



Σχήμα 6.5: Σύγκριση WER(%) και Ταχύτητας(it/s) ελέγχου για τις παραλλαγές του μοντέλου στο σύνολο δεδομένων Phoenix14.

Ένας πιο παραστατικός τρόπος για να απεικονιστούν οι συμβιβασμοί που πραγματοποιούν τα μοντέλα, παρουσιάζεται στο σχήμα 6.5. Πρόκειται για ένα διάγραμμα διασποράς που απεικονίζει τις ταχύτητες σε σχέση με τις αποδόσεις των μοντέλων, όπου ο αριθμός δίπλα από κάθε σημείο αντιπροσωπεύει το μέγεθος των εισόδων που εξετάζονται. Αυτό το διάγραμμα μπορεί να μας βοηθήσει να κάνουμε τις εξής

σημαντικές παρατηρήσεις:

1. Τα καλύτερα αποτελέσματα, τα παράγει το δίκτυο ResNet18, στα μεγέθη εισόδων 224 και 170.
2. Το δίκτυο ResNet18 παρουσιάζει τη μεγαλύτερη πτώση απόδοσης με τη μείωση των εισόδων
3. Το δίκτυο ShuffleNetv2 επιταχύνει σημαντικά την εκτέλεση του μοντέλου, αλλά έχει με διαφορά τις χειρότερες αποδόσεις.
4. Το δίκτυο SqueezeNet παρουσιάζει συστηματική πτώση απόδοσης με την μείωση του μεγέθους εισόδων, με μόνη εξαίρεση τη κατηγορία 150.
5. Το δίκτυο ShuffletNetv2 αυξάνει την απόδοση του όσο μειώνεται το μέγεθος των εισόδων
6. Το δίκτυο ResNet18 στην αρχή μειώνει σημαντικά την απόδοση του με τη μείωση του μεγέθους των εικόνων από 224 σε 190, αλλά με τη περαιτέρω μείωση σε 170, αυξάνει την απόδοση του.
7. Όλα τα μοντέλα αυξάνουν τη ταχύτητα τους όταν δέχονται μειωμένο μέγεθος εικονών στην είσοδο τους.

	Inputs	Test/Dev ($WER-WER_{original}$)	Test/Dev/Train Speed ($Speed - Speed_{original}$)	File size ($Size - Size_{original}$)	GFLOP ($GFLOP_{fwd} - GFLOP_{original+fwd}$)
ResNet18	224	22.3/21.2	4.09/4.23/-	411.8	1.81
	204	-	-	-	-0.21
	190	+9.6/+9.9	+1.3/+1.32/2.02	-0.1	-0.48
	170	+7.8/+8.6	+2.06/+2.2/2.35	-0.1	-0.66
	150	+7.8/+9.9	+3.36/+3.74/2.91	-0.1	-0.96
SqueezeNet1_1	224	-	-	-	-1.46
	204	+9.1/+9.4	+2.6/+2.73/2.97	-89.4	-1.52
	190	+10.1/+10.5	+3.39/+3.75/3.36	-89.4	-1.56
	170	+10.8/+11.5	+4.98/+5.61/3.78	-89.4	-1.61
	150	+10.6/+12.3	+7.2/+8.0/4.65	-89.4	-1.65
Shufflenetv2_x1_0	224	-	-	-	-1.67
	204	+13.5/+16.3	+3.83/+4.93/3.85	-87.6	-1.68
	190	+12.4/+13.9	+5.2/+5.9/4.52	-87.6	-1.71
	170	+12.1/+12.6	+7.02/+7.91/5.06	-87.6	-1.72
	150	+12.4/14.6	+10.82/+10.82/6.20	-87.6	-1.74

Σχήμα 6.6: Σύγκριση της απόδοσης, ταχύτητας, μεγέθους βαρών και GFLOP του αρχικού μοντέλου σε σχέση με τις παραλλαγές με τα SqueezeNet1_1, ShuffleNetv2_x1_0, ResNet18 και διαφορετικά μεγέθη εισόδων στο σύνολο Phoenix14. Για την ταχύτητα το μεγαλύτερο είναι καλύτερο, ενώ για τα υπόλοιπα το μεγαλύτερο είναι χειρότερο.

6.4 Έξοδοι μοντέλου

Στο ολοκλήρωμα κάθε εποχής, χρησιμοποιούνται τα βάρη που έχουν δημιουργηθεί μέχρι εκείνο το σημείο και το μοντέλο ελέγχεται στο σύνολο αξιολόγησης. Για την εξαγωγή στοιχείων σχετικά με αυτή την αξιολόγηση και την πιο ”όμορφη” παρουσίαση τους, χρησιμοποιείται το εργαλείο *sclite*, από το toolkit *kaldi*. Στα σχήματα 6.7, 6.8, 6.9 φαίνονται παραδείγματα από τις εξόδους με τη μορφοποίηση του sclite.

Στο σχήμα 6.7, βλέπουμε πως το μοντέλο έχει πραγματοποιήσει σωστά τις προβλέψεις. Οι επιπλέον πληροφορίες που παρέχονται είναι:

- Το id του διερμηνέα
- Τον φάκελο όπου βρίσκονται τα καρέ από την φράση που εξετάζεται
- Το αναλυτικό αποτέλεσμα, που φαίνεται στην γραμμή που ξεκινάει με τη λέξη SCORE. Περιλαμβάνει τον αριθμό των σωστών λέξεων που πρόβλεψε το μοντέλο (#Correct), τον αριθμό των λέξεων που αντικατέστησε (#Substitutions), τον αριθμό των λέξεων που διέγραψε (#Deletions) και τον αριθμό των λέξεων που πρόσθεσε (#Insertions). Παρακάτω θα αξιοποιηθεί επόμενο σχήμα για την παρουσίαση παραδείγματος.
- Η φράση αναφοράς, γνωστή και ως groundtruth.
- Ολόκληρη η φράση-πρόβλεψη του μοντέλου
- Η τελευταία σειρά αντιστοιχίζει το είδος λάθους του μοντέλου, στη τοποθεσία όπου το έκανε. Επειδή το μοντέλο δεν έκανε κανένα λάθος, η τελευταία σειρά είναι άδεια.

Αντίστοιχα, στο σχήμα 6.8 το μοντέλο έχει κάνει δύο διαγραφές, δηλαδή έχει τοποθετήσει τον κενό χαρακτήρα εκεί που κανονικά υπάρχει λέξη. Αυτό φαίνεται και από τη σειρά του SCORE, όπου έχει 3 σωστά και 2 διαγραφές, αλλά και από τη σειρά Eval, όπου το γράμμα D βρίσκεται κάτω από τις λέξεις που δεν υπάρχουν και υποδεικνύει τη διαγραφή. Τέλος, στο σχήμα 6.9 φαίνεται να έχει κάνει ένα λάθος διαγραφής και ένα λάθος αντικατάστασης, καθώς άλλαξε τη λέξη ”ΠΟΥ” από την αρχική φράση, με τη λέξη ”ΜΕΣΗΜΕΡΙ”.

```
10 Speaker sentences   0:  signer3   #utts: 588
11 id: (signer3-000)
12 File: health1_signer3_rep1_sentences-sentences0000
13 Channel: 1
14 Scores: (#C #S #D #I) 5 0 0 0
15 REF:  ΓΕΙΑ ΕΓΩ(1) ΜΠΟΡΩ ΒΟΗΘΩ ΠΩΣ
16 HYP:  ΓΕΙΑ ΕΓΩ(1) ΜΠΟΡΩ ΒΟΗΘΩ ΠΩΣ
17 Eval:
```

Σχήμα 6.7: Παράδειγμα πρόβλεψης όπου το μοντέλο δεν έκανε κανένα λάθος

Το σχήμα 6.10 βοηθάει επιπλέον στην κατανόηση των εξόδων των μοντέλων. Αυτό το σχήμα περιλαμβάνει μία φράση η οποία αποτελεί τη βασική αλήθεια και από κάτω φαίνονται οι προτάσεις-υποθέσεις του μοντέλου, εκπαιδευμένο στα διαφορετικά backbones. Από αυτά, μόνο το SqueezeNet έχει πετύχει

```
195 id: (signer3-023)
196 File: health1_signer3_rep2_sentences-sentences0009
197 Channel: 1
198 Scores: (#C #S #D #I) 3 0 2 0
199 REF:  ΠΡΕΠΕΙ ΕΣΥ ΒΙΒΛΙΟ ΥΓΕΙΑ ΕΧΩ
200 HYP:  ΠΡΕΠΕΙ ****** ΒΙΒΛΙΟ ΥΓΕΙΑ ******
201 Eval:             D                        D
202
```

Σχήμα 6.8: Παράδειγμα πρόβλεψης όπου το μοντέλο έκανε δύο λάθη διαγραφής (D)

```
274
275 id: (signer3-033)
276 File: health2_signer3_rep1_sentences-sentences0005
277 Channel: 1
278 Scores: (#C #S #D #I) 1 1 1 0
279 REF:  ΕΣΥ ΠΟΝΑΩ ΠΟΥ
280 HYP:  ΕΣΥ ********** ΜΕΣΗΜΕΡΙ
281 Eval:       D          S
282
```

Σχήμα 6.9: Παράδειγμα πρόβλεψης όπου το μοντέλο έκανε ένα λάθος διαγραφής (D) και ένα λάθος αντικατάστασης (S).

σωστά ολόκληρη τη φράση. Το ShuffleNetv2 έχει πραγματοποιήσει ένα λάθος διαγραφής, καθώς δεν κατάφερε να βρει τη λέξη ”ΕΓΩ(1)”, οπότε στη θέση της έβαλε τον κενό χαρακτήρα. Έπειτα, το ResNet18 διέπραξε ένα λάθος αντικατάστασης, καθώς μπέρδεψε τη λέξη ”ΒΟΗΘΩ” της αρχικής πρότασης, με τη λέξη ”ΕΞΥΠΗΡΕΤΩ”.

Τέλος, στο σχήμα 6.11, φαίνονται οι έξοδοι του μοντέλου χωρίς κάποια μορφοποίηση, όπως εξάγονται μετά την πραγματοποίηση ενός ελέγχου. Κάθε γλωσσική μονάδα αποτελεί μια ξεχωριστή γραμμή, ενώ η αντίστοιχη πρόταση φαίνεται από την πρώτη στήλη. Για παράδειγμα, η συγκεκριμένη εικόνα περιλαμβάνει τις εξόδους του μοντέλου για τις προτάσεις 0000-00013, του τρίτου διερμηνέα, από το πρώτο σύνολο προτάσεων που σχετίζονται με το νοσοκομείο.

Φράση	ΓΕΙΑ	ΕΓΩ(1)	ΜΠΟΡΩ	ΒΟΗΘΩ	ΠΩΣ
SqueezeNet1_1	ΓΕΙΑ	ΕΓΩ(1)	ΜΠΟΡΩ	ΒΟΗΘΩ	ΠΩΣ
ShuffleNetv2_x1_0	ΓΕΙΑ	******	ΜΠΟΡΩ	ΒΟΗΘΩ	ΠΩΣ
ResNet18	ΓΕΙΑ	ΕΓΩ(1)	ΜΠΟΡΩ	ΕΞΥΠΗΡΕΤΩ	ΠΩΣ

Σχήμα 6.10: Σύγκριση εξόδων μοντέλου με διαφορετικά backbones πάνω στο σύνολο GSL-SI

```
health1_signer3_rep1_sentences-sentences0000 1 0.00 0.01 ΓΕΙΑ
health1_signer3_rep1_sentences-sentences0000 1 0.01 0.02 ΕΓΩ(1)
health1_signer3_rep1_sentences-sentences0000 1 0.02 0.03 ΜΠΟΡΩ
health1_signer3_rep1_sentences-sentences0000 1 0.03 0.04 ΒΟΗΘΩ
health1_signer3_rep1_sentences-sentences0000 1 0.04 0.05 ΠΩΣ
health1_signer3_rep1_sentences-sentences0001 1 0.00 0.01 ΕΓΩ(1)
health1_signer3_rep1_sentences-sentences0001 1 0.01 0.02 ΧΡΕΙΑΖΟΜΑΙ
health1_signer3_rep1_sentences-sentences0001 1 0.02 0.03 ΓΙΑΤΡΟΣ(2)
health1_signer3_rep1_sentences-sentences0002 1 0.00 0.01 ΕΝΤΑΞΕΙ
health1_signer3_rep1_sentences-sentences0002 1 0.01 0.02 ΕΞΥΠΗΡΕΤΩ
health1_signer3_rep1_sentences-sentences0002 1 0.02 0.03 ΠΩΣ
health1_signer3_rep1_sentences-sentences0003 1 0.00 0.01 ΕΓΩ(1)
health1_signer3_rep1_sentences-sentences0003 1 0.01 0.02 ΖΑΛΙΖΟΜΑΙ
health1_signer3_rep1_sentences-sentences0003 1 0.02 0.03 ΣΥΝ
health1_signer3_rep1_sentences-sentences0003 1 0.03 0.04 ΠΟΛΥ
health1_signer3_rep1_sentences-sentences0003 1 0.04 0.05 ΝΕΡΟ
health1_signer3_rep1_sentences-sentences0004 1 0.00 0.01 ΕΣΥ
health1_signer3_rep1_sentences-sentences0004 1 0.01 0.02 ΦΑΡΜΑΚΟ
health1_signer3_rep1_sentences-sentences0004 1 0.02 0.03 ΧΑΠΙ
health1_signer3_rep1_sentences-sentences0004 1 0.03 0.04 ΠΑ
health1_signer3_rep1_sentences-sentences0005 1 0.00 0.01 ΟΧΙ
health1_signer3_rep1_sentences-sentences0006 1 0.00 0.01 ΠΡΩΤΟΝ
health1_signer3_rep1_sentences-sentences0006 1 0.01 0.02 ΕΣΥ
health1_signer3_rep1_sentences-sentences0006 1 0.02 0.03 ΡΑΝΤΕΒΟΥ
health1_signer3_rep1_sentences-sentences0006 1 0.03 0.04 ΚΡΑΤΗΣΗ
health1_signer3_rep1_sentences-sentences0006 1 0.04 0.05 ΓΙΑ
health1_signer3_rep1_sentences-sentences0006 1 0.05 0.06 ΓΙΑΤΡΟΣ(2)
health1_signer3_rep1_sentences-sentences0006 1 0.06 0.07 ΠΑΘΟΛΟΓΟΣ
health1_signer3_rep1_sentences-sentences0007 1 0.00 0.01 ΝΑΙ
health1_signer3_rep1_sentences-sentences0008 1 0.00 0.01 ΓΙΑΤΡΟΣ(2)
health1_signer3_rep1_sentences-sentences0008 1 0.01 0.02 ΕΝΤΟΛΗ
health1_signer3_rep1_sentences-sentences0008 1 0.02 0.03 ΙΔΙΟΚΤΗΤΗΣ
health1_signer3_rep1_sentences-sentences0009 1 0.00 0.01 ΠΡΕΠΕΙ
health1_signer3_rep1_sentences-sentences0009 1 0.01 0.02 ΒΙΒΛΙΟ
health1_signer3_rep1_sentences-sentences0009 1 0.02 0.03 ΥΓΕΙΑ
health1_signer3_rep1_sentences-sentences0009 1 0.03 0.04 ΕΧΩ
health1_signer3_rep1_sentences-sentences0010 1 0.00 0.01 ΕΓΩ(1)
health1_signer3_rep1_sentences-sentences0010 1 0.01 0.02 ΘΕΛΩ
health1_signer3_rep1_sentences-sentences0010 1 0.02 0.03 ΣΥΝ
health1_signer3_rep1_sentences-sentences0010 1 0.03 0.04 ΑΙΜΑ
health1_signer3_rep1_sentences-sentences0010 1 0.04 0.05 ΕΞΕΤΑΣΗ
health1_signer3_rep1_sentences-sentences0011 1 0.00 0.01 ΓΙΑΤΡΟΣ(2)
health1_signer3_rep1_sentences-sentences0011 1 0.01 0.02 ΒΙΒΛΙΟ
health1_signer3_rep1_sentences-sentences0011 1 0.02 0.03 ΓΡΑΦΩ
health1_signer3_rep1_sentences-sentences0011 1 0.03 0.04 ΑΙΜΑ
health1_signer3_rep1_sentences-sentences0011 1 0.04 0.05 ΕΞΕΤΑΣΗ
health1_signer3_rep1_sentences-sentences0012 1 0.00 0.01 ΕΝΤΑΞΕΙ
health1_signer3_rep1_sentences-sentences0012 1 0.01 0.02 ΑΠΟΓΕΥΜΑ
health1_signer3_rep1_sentences-sentences0013 1 0.00 0.01 ΚΑΛΟ
health1_signer3_rep1_sentences-sentences0013 1 0.01 0.02 ΑΠΟΓΕΥΜΑ
```

Σχήμα 6.11: Προβλέψεις μοντέλου πάνω στο σύνολο GSL-SI

Κεφάλαιο 7ο: Συζήτηση

7.1 Ανάλυση Αποτελεσμάτων

Κάτι που πρέπει να τονιστεί για το σύνολο GSL, είναι πως με τον διαχωρισμό train/dev/test του SI, το σύνολο ελέγχου, δεν περιλαμβάνει καμία άγνωστη φράση, απλά περιλαμβάνει έναν διαφορετικό διερμηνέα να την εκτελεί. Αυτό σημαίνει πως αυτό που καλείται να κάνει το μοντέλο στη προκειμένη περίπτωση, είναι να αναγνωρίσει μία ήδη γνωστή φράση, προσπερνώντας μόνο ιδιωματισμούς και χαρακτηριστικά που σχετίζονται με την εμφάνιση των διερμηνέων. Είναι, λοιπόν, σχετικά παράδοξο που τόσα μοντέλα παρουσιάζουν τόση δυσκολία σε αυτή τη διεργασία. Παρ'όλα αυτά, τα μοντέλα που εξετάστηκαν, έχουν επιτύχει παρόμοια, αν όχι μικρότερη απόκλιση, σε σχέση με άλλα σύνολα. Υπάρχουν όμως και δύο μοντέλα τα οποία παρουσίασαν εξαιρετικά καλά αποτελέσματα. Σχετικά με το GoogLeNet+TConvs [52], πιστεύεται πως οφείλεται στη χρήση της τεχνικής iteration training, το οποία έχει αποδειχθεί πως βοηθάει δραστικά εναντίον του overfitting, αυξάνοντας όμως σημαντικά τις υπολογιστικές απαιτήσεις. Επιπλέον, στο I3DBLSTM [15], οι συγγραφείς πραγματοποιούν εξειδικευμένη προεκπαίδευση στο μοντέλο CNN που χρησιμοποιούν.

Επίσης, ένα γενικό συμπέρασμα που μπορεί να εξαχθεί από τα δεδομένα, τόσο αυτά που παρουσιάστηκαν σε αυτή την εργασία, όσο και από το άρθρο των συγγραφέων, είναι πως το σύνολο δεδομένων GSL, "ωθεί" τα μοντέλα προς το overfitting. Παρά το γεγονός πως οι προτάσεις είναι όμοιες σε αριθμό με αυτές του συνόλου Phoenix14, ο πολύ μικρός αριθμός διαφορετικών γλωσσικών μονάδων που χρησιμοποιείται, καθώς και ο μικρός αριθμός διαφορετικών προτάσεων, περιορίζει την ικανότητα του μοντέλου να "μάθει" τη γλώσσα. Αυτά τα μεγέθη εικάζεται πως δεν είναι αρκετά για να εκπαιδευτεί ικανοποιητικά ένα μοντέλο βαθιάς μάθησης το οποίο θα μπορεί να πραγματοποιήσει ικανοποιητικές προβλέψεις σε άγνωστα δεδομένα. Μπορεί όμως, πιθανώς, να χρησιμοποιηθεί ώστε ένα μοντέλο να μάθει αυτόν τον περιορισμένο αριθμό φράσεων και γλωσσικών μονάδων και να μπορεί να τις αναγνωρίσει.

Όσον αφορά το κομμάτι της έρευνας που αφορούσε τη μείωση των μεγεθών των εισόδων, η οποία ήταν αυτή που παρήγαγε τα πιο ενδιαφέροντα αποτελέσματα, μπορεί να χωριστεί σε δύο κομμάτια. Το πρώτο αφορά τον σχολιασμό της ταχύτητας και του υπολογιστικού κόστους και το επόμενο τις αποδόσεις.

Η σμίκρυνση των εισόδων επιφέρει μια αναμενόμενη αύξηση ταχύτητας, η οποία όπως φαίνεται και από τα σχήματα 6.3, 6.4, δεν εξαρτάται μόνο από τις υπολογιστικές απαιτήσεις, αλλά και από άλλες παραμέτρους, ικανοποιώντας την αρχική πρόβλεψη. Παρ' όλα αυτά, οι λιγότερες πράξεις που πρέπει να πραγματοποιηθούν, ιδίως στα συνελικτικά στρώματα μεταξύ των, σμικρυσμένων, χαρτών χαρακτηριστικών και των φίλτρων σίγουρα διαδραματίζουν σημαντικό ρόλο, όπως επίσης φαίνεται από την συσχέτιση των σχημάτων. Άλλη μια παράμετρος, πιθανόν είναι η ανάγκη για φόρτωση λιγότερων δεδομένων στην μνήμη, που συνεπάγεται την επιτάχυνση της διαδικασίας.

Το δεύτερο κομμάτι της έρευνας σχετικά με τις αποδόσεις των διαφορετικών μεγεθών εισόδων, ήταν αυτό που παρήγαγε τα πιο ιδιαίτερα αποτελέσματα. Η αρχική υπόθεση ήταν πως η απόδοση θα μειώνεται όσο μειώνεται το μέγεθος των εικόνων στην είσοδο. Η μείωση μπορεί να ήταν πιο μικρή και μετά να γινόταν μεγάλη, απότομα, επειδή κάποια στιγμή το μοντέλο θα έχανε πολύ σημαντική πληροφορία. Αντίθετα, τα αποτελέσματα για δύο από τα τρία backbones που χρησιμοποιήθηκαν ίσχυε κάτι διαφορετικό. Το

ShuffleNetv2 βελτιώθηκε συστηματικά όσο μειωνόταν το μέγεθος, ενώ το ResNet18 έκανε μία πολύ έντονη πτώση (+9.6% WER) από το μέγεθος 224 στο 190, αλλά στο μέγεθος 170, βελτιώθηκε (κατά 1.8% WER). Μόνο το SqueezeNet ακολούθησε την αρχική υπόθεση και μειώθηκε κατά περίπου 1% σε κάθε μείωση μεγέθους. Οι αυξήσεις και οι μειώσεις είναι μικρές, οπότε θα μπορούσαν να αποδοθούν σε τυχαίες διακυμάνσεις κατά την εκπαίδευση ή στην αρχικοποίηση των βαρών. Ακόμα και σε αυτή τη περίπτωση όμως, τα δεδομένα υποδηλώνουν πως τα μοντέλα έχουν ένα ”περιθώριο” μείωσης εισόδων κατά το οποίο διατηρούν την απόδοση τους, ενώ η ταχύτητα επεξεργασίας και το υπολογιστικό κόστος συνεχίζονται να μειώνονται με τους ίδιους ρυθμούς.

Η αιτία αυτού του φαινομένου δεν είναι ξεκάθαρη, καθώς η περικοπή εικόνων από 256×256 σε 150×150 είναι πολύ σημαντική και σίγουρα αρκετή πληροφορία περικόπτεται. Πιθανόν όμως υποδεικνύει πως το σύνολο δεδομένων περιέχει έναν αρκετά καλό αριθμό επαναλήψεων των φράσεων και το μοντέλο επιτυγχάνει να εξάγει ουσιαστικά χαρακτηριστικά. Μία πιο συγκεκριμένη υπόθεση, είναι πως με την περικοπή τον εικόνων, το μοντέλο επικεντρώνεται σε πληροφορία που αφορούν κυρίως τις κινήσεις των χεριών και του στόματος, οι οποίες όμως περιλαμβάνουν αρκετή πληροφορία για να διατηρήσουν το μοντέλο στις ίδιες αποδόσεις, μέχρι η περικοπή να γίνει αρκετά μεγάλη, ώστε να μην μπορούν να εξαχθούν κατάλληλα χαρακτηριστικά και από αυτά. Αυτή την υπόθεση την υποστηρίζει και μία επιπλέον μεμονωμένη δοκιμή που πραγματοποιήθηκε με το ResNet18 και 100 εισόδους, όπου η απόδοση έπεσε πολύ έντονα, γύρω στο 60% στο σύνολο αξιολόγησης. Σε κάθε περίπτωση, τα δεδομένα είναι πολύ ενθαρρυντικά και δείχνουν πως τα μοντέλα μπορούν να λειτουργήσουν και με μικρές εισόδους, εάν η πληροφορία που περιέχουν είναι κατάλληλη. Με περαιτέρω έρευνα και διασταύρωση θα μπορούσαν να αποτελέσουν έναν πολύ ικανοποιητικό και εύκολο τρόπο μείωσης πολυπλοκότητας και αύξησης ταχύτητας των μοντέλων.

7.2 Δυσκολίες που αντιμετωπίστηκαν

7.2.0.1 Αρχική Επιλογή Μοντέλου

- **Ρυθμίσεις Περιβάλλοντος**: Η δημιουργία του αρχικού εικονικού περιβάλλοντος και η εγκατάσταση των απαραίτητων βιβλιοθηκών ήταν μια μεγάλη πρόκληση. Στα πρώτα μοντέλα που δοκιμάστηκαν, πολλές από τις βιβλιοθήκες που θεωρούνταν αναγκαίες, δημιουργούσαν προβλήματα με ασυμβατότητες μεταξύ τους, ή ήταν παρωχημένες και άλλες εκδόσεις δεν ταίριαζαν κατάλληλα. Επίσης, ορισμένες βιβλιοθήκες, όπως το **ctcdecode** που χρησιμοποιούταν από τα περισσότερα μοντέλα για την υλοποίηση της συνάρτησης απώλειας CTC, αντιμετώπιζαν συχνά προβλήματα με το λειτουργικό σύστημα Windows, με αποτέλεσμα να πρέπει να γίνει εγκατάσταση Linux. Τέλος, έπρεπε να γίνουν δοκιμές ώστε να επιλεχθεί μια κατάλληλη έκδοση PyTorch που να ικανοποιεί όλες τις βιβλιοθήκες που χρησιμοποιούνταν.
- **CorrNet+**: Το CorrNet+ ήταν το πρώτο μοντέλο που έτρεξε επιτυχώς, αλλά πέρα από τις δυσκολίες που αντιμετωπίστηκαν για τη δημιουργία του περιβάλλοντος, η αρχική επιτυχία δεν ήταν πλήρης γιατί παρόλο που το μοντέλο πραγματοποίησε τον έλεγχο, με βάρη τα οποία παρέχουν οι συγγραφείς, τα αποτελέσματα ήταν αρκετά υποδεέστερα από αυτά που υπήρχαν στο συγκεκριμένο άρθρο. Για να επιλυθεί το πρόβλημα, έγινε επικοινωνία με τους συγγραφείς μέσω των ”Issues” της πλατφόρμας GitHub, αλλά δε μπόρεσε να βρεθεί κάποια λύση, οπότε το μοντέλο δε μπορούσε να χρησιμοποιηθεί.

- **Αποσφαλμάτωση**: Λόγω του τρόπου με τον οποίο λειτουργεί το μοντέλο, κατά την εκπαίδευση φορτώνονται δεδομένα στην κάρτα γραφικών για να επεξεργαστούν από τη βιβλιοθήκη PyTorch αξιοποιώντας τις ικανότητες της για παραλληλισμό. Αυτή όμως η μετάβαση, εικάζεται πως προκαλούσε πρόβλημα διαχείρισης στον αποσφαλματωτή του προγράμματος επεξεργασίας κώδικα που χρησιμοποιήθηκε, με αποτέλεσμα σε κάθε προσπάθεια για αποσφαλμάτωση κατά την εκπαίδευση, να παγώνει το σύστημα και να μην ανταποκρίνεται. Καθώς η αδυναμία αποσφαλμάτωσης είναι πάρα πολύ αρνητικό για ένα εκτενές πρόγραμμα, όπως αυτό του μοντέλου, αυτό αποτέλεσε μεγάλο πρόβλημα. Η λύση που χρησιμοποιήθηκε τελικά ήταν η δημιουργία ενός αντιγράφου του μοντέλου, το οποίο όμως δεν φόρτωνε τα δεδομένα στη κάρτα γραφικών, αλλά στον επεξεργαστή.

- **Χωρητικότητα**: Ο μεγάλος όγκος των συνόλων δεδομένων αποτέλεσε ένα ακόμα πρόβλημα, καθώς κάθε σύνολο είχε μέγεθος περίπου 50GB και μετά την προεπεξεργασία που δημιουργούνταν και αντίγραφα των εικόνων, ο όγκος μεγάλωνε περαιτέρω. Για να αποφευχθεί το αχρείαστο οικονομικό κόστος της αποθήκευσης στο cloud και πιθανώς η αύξηση του χρόνου που χρειαζόταν για τις διαδικασίες, χρησιμοποιήθηκε ένας επιπλέον δίσκος SSD.

- **Αδυναμία χρήσης διαδικτυακών πλατφορμών εκπαίδευσης**: Στην αρχή εξετάστηκε το ενδεχόμενο να χρησιμοποιηθούν πλατφόρμες όπως το Google Collab και Kaggle, οι οποίες παρέχουν και κάρτες γραφικών για την επιτάχυνση της εκπαίδευσης. Δεν βρέθηκε όμως τρόπος να εγκατασταθούν απαραίτητες βιβλιοθήκες (όπως το ctcdecode).

- **Διάρκεια εκπαιδεύσεων**: Ένα από τα μεγαλύτερα προβλήματα που αντιμετωπίστηκε και αποτέλεσε και λόγο έλλειψης κάποιων επιπλέον αποτελεσμάτων, για παράδειγμα στο σύνολο GSL, ήταν πως κάθε εκπαίδευση απαιτούσε περίπου μία μέρα για την ολοκλήρωση της. Αυτό σήμαινε πως από τη στιγμή που αποφασίστηκε το πως θα τρέξουν τα μοντέλα και με τι παραμέτρους, χρειάστηκαν περίπου 20 μέρες συνεχών εκπαιδεύσεων μόνο για τη λήψη των αποτελεσμάτων.

Κεφάλαιο 8ο: Συμπεράσματα ή/και προτάσεις βελτίωσης

Συμπερασματικά σε αυτή την εργασία, μελετήθηκαν μοντέλο με σκοπό την αναγνώριση της νοηματικής γλώσσας. Το VAC_CSLR επιλέχθηκε για την καλή υλοποίηση του και τις μικρές σχετικά υπολογιστικές του απαιτήσεις σε συνδυασμό με καλά αποτελέσματα. Εξετάστηκαν παραλλαγές του, χρησιμοποιώντας δύο επιπλέον CNN, τα SqueezeNet1_1 και ShuffleNetv2_x1_0. Αυτές οι παραλλαγές δοκιμάστηκαν στο σύνολο δεδομένων *Greek Sign Language Dataset* με σκοπό την εξέταση της δυνατότητας αξιοποίησης για πραγματικές συνθήκες, όπου τα αποτελέσματα υπέδειξαν πως χρειάζονται αλλαγές προτού να μπορέσει να χρησιμοποιηθεί για τέτοιο σκοπό, αλλά θα μπορούσε να αξιοποιηθεί για ερευνητική μελέτη. Επιπλέον, δοκιμάστηκε στο σύνολο δεδομένων Phoenix14, όπου το μοντέλο με SqueezeNet πέτυχε την καλύτερη απόδοση, σε περιβάλλον με περιορισμένους υπολογιστικούς πόρους, ρίχνοντας την απόδοση αλλά προσφέροντας επιτάχυνση. Τέλος, πραγματοποιήθηκε μία επιπλέον έρευνα για τις αποδόσεις των παραλλαγών του μοντέλου με μειωμένο αριθμό εισόδων, περικόπτωντας περαιτέρω τις εικόνες εισόδου. Τα αποτελέσματα σε αυτό το σημείο, υπέδειξαν πως υπάρχει μια αρχική πτώση, αλλά στη συνέχεια η απόδοση παραμένει περίπου σταθερή, ενώ παράλληλα η ταχύτητα επεξεργασίας του μοντέλου αυξάνεται σημαντικά και το υπολογιστικό κόστος μειώνεται. Συγκεκριμένα, επιτεύχθηκε, για αυτό το μοντέλο τουλάχιστον, ο σχεδόν διπλασιασμός της ταχύτητας και ο υποδιπλασιασμός του υπολογιστικού κόστους με μείωση της απόδοσης ίση με 7.8% χρησιμοποιώντας το ResNet18 με μέγεθος εισόδων 150×150.

Κάτι που θα ήταν ενδιαφέρον να πραγματοποιηθεί για να διασταυρώσει και να προσθέσει βαρύτητα στην ικανότητα γενίκευσης των αποτελεσμάτων, είναι να γίνει παρόμοια δοκιμή με μείωση των εισόδων σε άλλα σύνολα δεδομένων και με άλλα μοντέλα.

Έπειτα, θα άξιζε να διερευνηθεί η χρήση βέλτιστων μεθόδων για μείωση του μεγέθους των εικόνων με ταυτόχρονη διατήρηση κρίσιμης πληροφορίας. Με αυτόν τον τρόπο θα μπορούσε να επιτευχθεί η επιθυμητή μείωση χωρίς απώλεια σημαντικών χαρακτηριστικών, τα οποία μπορεί να χάνονται με απλές τεχνικές, όπως η περικοπή, αλλά είναι απαραίτητα για την εκμάθηση του μοντέλου. Τέλος, κάτι που πρέπει να μελετηθεί είναι η δυνατότητα του μοντέλου να παρέχει προβλέψεις σε πραγματικό χρόνο και όχι μόνο με είσοδο ολοκληρωμένες φράσεις.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] N. Geographic, *The Creation of Sign Language*. διεύθν.: https://www.nationalgeographic.com/history/history-magazine/article/creation-of-sign-language.

[2] W. C. Stokoe, "Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf", en, *Journal of Deaf Studies and Deaf Education*, τόμ. 10, αρθμ. 1, σσ. 3–37, Ιαν. 2005, issn: 1465-7325. doi: 10.1093/deafed/eni001. διεύθν.: https://academic.oup.com/jdsde/article-lookup/doi/10.1093/deafed/eni001.

[3] W. F. of the Deaf, *World Federation of the Deaf*. διεύθν.: https://wfdeaf.org/.

[4] Z. Liang, H. Li και J. Chai, "Sign Language Translation: A Survey of Approaches and Techniques", en, *Electronics*, τόμ. 12, αρθμ. 12, σ. 2678, Ιούν. 2023, issn: 2079-9292. doi: 10.3390/electronics12122678. διεύθν.: https://www.mdpi.com/2079-9292/12/12/2678.

[5] T. Starner και A. Pentland, "Real-time American Sign Language recognition from video using hidden Markov models", στο *Proceedings of International Symposium on Computer Vision - ISCV*, Coral Gables, FL, USA: IEEE Comput. Soc. Press, 1995, σσ. 265–270, isbn: 978-0-8186-7190-6. doi: 10.1109/ISCV.1995.477012. διεύθν.: http://ieeexplore.ieee.org/document/477012/.

[6] S. Mehdi και Y. Khan, "Sign language recognition using sensor gloves", στο *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02.*, Singapore: IEEE, 2002, 2204–2206 vol.5, isbn: 978-981-04-7524-6. doi: 10.1109/ICONIP.2002.1201884. διεύθν.: http://ieeexplore.ieee.org/document/1201884/.

[7] N. Pugeault και R. Bowden, "Spelling it out: Real-time ASL fingerspelling recognition", στο *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, Barcelona, Spain: IEEE, Νοέ. 2011, σσ. 1114–1119, isbn: 978-1-4673-0063-6. doi: 10.1109/ICCVW.2011.6130290. διεύθν.: http://ieeexplore.ieee.org/document/6130290/.

[8] L. Pigou, S. Dieleman, P.-J. Kindermans και B. Schrauwen, "Sign Language Recognition Using Convolutional Neural Networks", en, στο *Computer Vision - ECCV 2014 Workshops*, L. Agapito, M. M. Bronstein και C. Rother, επιμελητές, τόμ. 8925, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2015, σσ. 572–578, isbn: 978-3-319-16177-8. doi: 10.1007/978-3-319-16178-5_40. διεύθν.: http://link.springer.com/10.1007/978-3-319-16178-5_40.

[9] D. Li, C. R. Opazo, X. Yu και H. Li, *Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison*, Version Number: 2, 2019. doi: 10.48550/ARXIV.1910.11006. διεύθν.: https://arxiv.org/abs/1910.11006.

[10] C. Vogler και D. Metaxas, "Adapting Hidden Markov Models for ASL recognition by using three-dimensional computer vision methods", English (US), *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, τόμ. 1, σσ. 156–161, 1997, Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. Part 1 (of 5) ; Conference date: 12-10-1997 Through 15-10-1997, issn: 0884-3627.

[11] C. Vogler και D. Metaxas, “Parallel hidden Markov models for American sign language recognition”, στο *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, Greece: IEEE, 1999, 116–122 vol.1, isbn: 978-0-7695-0164-2. doi: 10.1109/ICCV.1999.791206. διεύθν.: http://ieeexplore.ieee.org/document/791206/.

[12] W. Gao, G. Fang, D. Zhao και Y. Chen, “A Chinese sign language recognition system based on SOFM/SRN/HMM”, en, *Pattern Recognition*, τόμ. 37, αρθμ. 12, σσ. 2389–2402, Δεκ. 2004, issn: 00313203. doi: 10.1016/S0031-3203(04)00165-7. διεύθν.: https://linkinghub.elsevier.com/retrieve/pii/S0031320304001657.

[13] J. Han, G. Awad και A. Sutherland, “Modelling and segmenting subunits for sign language recognition based on hand motion analysis”, en, *Pattern Recognition Letters*, τόμ. 30, αρθμ. 6, σσ. 623–633, Απρ. 2009, issn: 01678655. doi: 10.1016/j.patrec.2008.12.010. διεύθν.: https://linkinghub.elsevier.com/retrieve/pii/S0167865509000087.

[14] O. Koller, N. C. Camgoz, H. Ney και R. Bowden, “Weakly Supervised Learning with Multi-Stream CNN-LSTM-HMMs to Discover Sequential Parallelism in Sign Language Videos”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, τόμ. 42, αρθμ. 9, σσ. 2306–2320, Σεπτ. 2020, issn: 0162-8828, 2160-9292, 1939-3539. doi: 10.1109/TPAMI.2019.2911077. διεύθν.: https://ieeexplore.ieee.org/document/8691602/.

[15] J. Carreira και A. Zisserman, *Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset*, Version Number: 3, 2017. doi: 10.48550/ARXIV.1705.07750. διεύθν.: https://arxiv.org/abs/1705.07750.

[16] Y. Min, A. Hao, X. Chai και X. Chen, *Visual Alignment Constraint for Continuous Sign Language Recognition*, Version Number: 2, 2021. doi: 10.48550/ARXIV.2104.02330. διεύθν.: https://arxiv.org/abs/2104.02330.

[17] L. Hu, W. Feng, L. Gao, Z. Liu και L. Wan, *CorrNet+: Sign Language Recognition and Translation via Spatial-Temporal Correlation*, Version Number: 1, 2024. doi: 10.48550/ARXIV.2404.11111. διεύθν.: https://arxiv.org/abs/2404.11111.

[18] J. Ahn, Y. Jang και J. S. Chung, “Slowfast Network for Continuous Sign Language Recognition”, στο *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Seoul, Korea, Republic of: IEEE, Απρ. 2024, σσ. 3920–3924, isbn: 979-8-3503-4485-1. doi: 10.1109/ICASSP48485.2024.10445841. διεύθν.: https://ieeexplore.ieee.org/document/10445841/.

[19] L. Hu, L. Gao, Z. liu και W. Feng, *Self-Emphasizing Network for Continuous Sign Language Recognition*, Version Number: 1, 2022. doi: 10.48550/ARXIV.2211.17081. διεύθν.: https://arxiv.org/abs/2211.17081.

[20] H. Zhou, W. Zhou, Y. Zhou και H. Li, *Spatial-Temporal Multi-Cue Network for Continuous Sign Language Recognition*, Version Number: 1, 2020. doi: 10.48550/ARXIV.2002.03187. διεύθν.: https://arxiv.org/abs/2002.03187.

[21] Y. Chen, R. Zuo, F. Wei, Y. Wu, S. Liu και B. Mak, *Two-Stream Network for Sign Language Recognition and Translation*, Version Number: 2, 2022. doi: 10.48550/ARXIV.2211.01367. διεύθν.: https://arxiv.org/abs/2211.01367.

[22] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney και R. Bowden, "Neural Sign Language Translation", στο *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT: IEEE, Ιούν. 2018, σσ. 7784–7793, isbn: 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00812. διεύθν.: https://ieeexplore.ieee.org/document/8578910/.

[23] N. C. Camgoz, O. Koller, S. Hadfield και R. Bowden, *Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation*, Version Number: 1, 2020. doi: 10.48550/ARXIV.2003.13830. διεύθν.: https://arxiv.org/abs/2003.13830.

[24] B. Zhang, M. Müller και R. Sennrich, *SLTUNET: A Simple Unified Model for Sign Language Translation*, Version Number: 1, 2023. doi: 10.48550/ARXIV.2305.01778. διεύθν.: https://arxiv.org/abs/2305.01778.

[25] O. Koller, J. Forster και H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers", en, *Computer Vision and Image Understanding*, τόμ. 141, σσ. 108–125, Δεκ. 2015, issn: 10773142. doi: 10.1016/j.cviu.2015.09.013. διεύθν.: https://linkinghub.elsevier.com/retrieve/pii/S1077314215002088.

[26] N. Adaloglou, T. Chatzis, I. Papastratis κ.ά., "A Comprehensive Study on Deep Learning-Based Methods for Sign Language Recognition", *IEEE Transactions on Multimedia*, τόμ. 24, σσ. 1750–1762, 2022, issn: 1520-9210, 1941-0077. doi: 10.1109/TMM.2021.3070438. διεύθν.: https://ieeexplore.ieee.org/document/9393618/.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li και Li Fei-Fei, "ImageNet: A large-scale hierarchical image database", στο *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL: IEEE, Ιούν. 2009, σσ. 248–255, isbn: 978-1-4244-3992-8. doi: 10.1109/CVPR.2009.5206848. διεύθν.: https://ieeexplore.ieee.org/document/5206848/.

[28] T. M. Mitchell, *Machine learning* (McGraw-Hill series in Computer Science), eng, Nachdr. New York: McGraw-Hill, 2013, isbn: 978-0-07-042807-2.

[29] C. Henley, *Foundations of Neuroscience* (Open textbook library). Michigan State University, 2021. διεύθν.: https://books.google.gr/books?id=rRCKzgEACAAJ.

[30] W. S. McCulloch και W. Pitts, "A logical calculus of the ideas immanent in nervous activity", en, *The Bulletin of Mathematical Biophysics*, τόμ. 5, αρθμ. 4, σσ. 115–133, Δεκ. 1943, issn: 0007-4985, 1522-9602. doi: 10.1007/BF02478259. διεύθν.: http://link.springer.com/10.1007/BF02478259.

[31] G. Obaido, I. D. Mienye, O. F. Egbelowo κ.ά., "Supervised machine learning in drug discovery and development: Algorithms, applications, challenges, and prospects", en, *Machine Learning with Applications*, τόμ. 17, σ. 100 576, Σεπτ. 2024, issn: 26668270. doi: 10.1016/j.mlwa.2024.100576. διεύθν.: https://linkinghub.elsevier.com/retrieve/pii/S2666827024000525.

[32] M. Minsky και S. Papert, *Perceptrons: an introduction to computational geometry*, Expanded ed. Cambridge, Mass: MIT Press, 1988, isbn: 978-0-262-63111-2.

[33] Y. Lecun, L. Jackel, C. Cortes κ.ά., "Learning Algorithms For Classification: A Comparison On Handwritten Digit Recognition", *The Statistical Mechanics Perspective*, Ιούλ. 2000.

[34] D. H. Hubel και T. N. Wiesel, “Receptive fields of single neurones in the cat’s striate cortex”, en, *The Journal of Physiology*, τόμ. 148, αρθμ. 3, σσ. 574–591, Οκτ. 1959, issn: 0022-3751, 1469-7793. doi: 10.1113/jphysiol.1959.sp006308. διεύθν.: https://physoc.onlinelibrary.wiley.com/doi/10.1113/jphysiol.1959.sp006308.

[35] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems*, eng, Second edition. Beijing Boston Farnham Sebastopol Tokyo: O’Reilly, 2019, isbn: 978-1-4920-3264-9.

[36] H. Wu και X. Gu, “Max-Pooling Dropout for Regularization of Convolutional Neural Networks”, στο *Neural Information Processing*, S. Arik, T. Huang, W. K. Lai και Q. Liu, επιμελητές, τόμ. 9489, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2015, σσ. 46–54, isbn: 978-3-319-26531-5. doi: 10.1007/978-3-319-26532-2_6. διεύθν.: http://link.springer.com/10.1007/978-3-319-26532-2_6.

[37] K. He, X. Zhang, S. Ren και J. Sun, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition”, en, στο *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele και T. Tuytelaars, επιμελητές, τόμ. 8691, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2014, σσ. 346–361, isbn: 978-3-319-10577-2. doi: 10.1007/978-3-319-10578-9_23. διεύθν.: http://link.springer.com/10.1007/978-3-319-10578-9_23.

[38] W. Ouyang, X. Wang, X. Zeng κ.ά., “DeepID-Net: Deformable deep convolutional neural networks for object detection”, στο *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, Ιούν. 2015, σσ. 2403–2412, isbn: 978-1-4673-6964-0. doi: 10.1109/CVPR.2015.7298854. διεύθν.: http://ieeexplore.ieee.org/document/7298854/.

[39] D. Hutchison, T. Kanade, J. Kittler κ.ά., “Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition”, στο *Artificial Neural Networks – ICANN 2010*, K. Diamantaras, W. Duch και L. S. Iliadis, επιμελητές, τόμ. 6354, Series Title: Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, σσ. 92–101, isbn: 978-3-642-15824-7. doi: 10.1007/978-3-642-15825-4_10. διεύθν.: http://link.springer.com/10.1007/978-3-642-15825-4_10.

[40] J. Cardete, *Convolutional Neural Networks: A Comprehensive Guide*, 2024. διεύθν.: https://medium.com/thedeephub/convolutional-neural-networks-a-comprehensive-guide-5cc0b5eae175.

[41] C. Szegedy, W. Liu, Y. Jia κ.ά., *Going Deeper with Convolutions*, Version Number: 1, 2014. doi: 10.48550/ARXIV.1409.4842. διεύθν.: https://arxiv.org/abs/1409.4842.

[42] K. He, X. Zhang, S. Ren και J. Sun, *Deep Residual Learning for Image Recognition*, Version Number: 1, 2015. doi: 10.48550/ARXIV.1512.03385. διεύθν.: https://arxiv.org/abs/1512.03385.

[43] F. Ramzan, M. U. G. Khan, A. Rehmat κ.ά., “A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer’s Disease Stages Using Resting-State fMRI and Residual Neural Networks”, en, *Journal of Medical Systems*, τόμ. 44, αρθμ. 2, σ. 37, Φεβ. 2020, issn: 0148-5598, 1573-689X. doi: 10.1007/s10916-019-1475-2. διεύθν.: http://link.springer.com/10.1007/s10916-019-1475-2.

[44] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally και K. Keutzer, *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size*, Version Number: 4, 2016. doi: 10.48550/ARXIV.1602.07360. διεύθν.: https://arxiv.org/abs/1602.07360.

[45] N. Ma, X. Zhang, H.-T. Zheng και J. Sun, *ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design*, Version Number: 1, 2018. doi: 10.48550/ARXIV.1807.11164. διεύθν.: https://arxiv.org/abs/1807.11164.

[46] X. Zhang, X. Zhou, M. Lin και J. Sun, *ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices*, Version Number: 2, 2017. doi: 10.48550/ARXIV.1707.01083. διεύθν.: https://arxiv.org/abs/1707.01083.

[47] S. Hochreiter και J. Schmidhuber, "Long Short-Term Memory", en, *Neural Computation*, τόμ. 9, αρθμ. 8, σσ. 1735–1780, Νοέ. 1997, issn: 0899-7667, 1530-888X. doi: 10.1162/neco.1997.9.8.1735. διεύθν.: https://direct.mit.edu/neco/article/9/8/1735-1780/6109.

[48] A. Graves και J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures", en, *Neural Networks*, τόμ. 18, αρθμ. 5-6, σσ. 602–610, Ιούλ. 2005, issn: 08936080. doi: 10.1016/j.neunet.2005.06.042. διεύθν.: https://linkinghub.elsevier.com/retrieve/pii/S0893608005001206.

[49] A. Graves, S. Fernández, F. Gomez και J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks", en, στο *Proceedings of the 23rd international conference on Machine learning - ICML '06*, Pittsburgh, Pennsylvania: ACM Press, 2006, σσ. 369–376, isbn: 978-1-59593-383-6. doi: 10.1145/1143844.1143891. διεύθν.: http://portal.acm.org/citation.cfm?doid=1143844.1143891.

[50] R. Zuo, F. Wei και B. Mak, *Towards Online Continuous Sign Language Recognition and Translation*, Version Number: 2, 2024. doi: 10.48550/ARXIV.2401.05336. διεύθν.: https://arxiv.org/abs/2401.05336.

[51] PyTorch, *Torchvision Models*, Accessed: 2025-01-18, 2025. διεύθν.: https://pytorch.org/vision/main/models.html.

[52] R. Cui, H. Liu και C. Zhang, "A Deep Neural Framework for Continuous Sign Language Recognition by Iterative Training", *IEEE Transactions on Multimedia*, τόμ. 21, αρθμ. 7, σσ. 1880–1891, Ιούλ. 2019, issn: 1520-9210, 1941-0077. doi: 10.1109/TMM.2018.2889563. διεύθν.: https://ieeexplore.ieee.org/document/8598757/.

[53] J. Kan, K. Hu, M. Hagenbuchner, A. C. Tsoi, M. Bennamounm και Z. Wang, *Sign Language Translation with Hierarchical Spatio-TemporalGraph Neural Network*, Version Number: 1, 2021. doi: 10.48550/ARXIV.2111.07258. διεύθν.: https://arxiv.org/abs/2111.07258.

[54] B. Natarajan, E. Rajalakshmi, R. Elakkiya κ.ά., "Development of an End-to-End Deep Learning Framework for Sign Language Recognition, Translation, and Video Generation", *IEEE Access*, τόμ. 10, σσ. 104 358–104 374, 2022, issn: 2169-3536. doi: 10.1109/ACCESS.2022.3210543. διεύθν.: https://ieeexplore.ieee.org/document/9905589/.

[55] L. Hu, L. Gao, Z. Liu και W. Feng, *Continuous Sign Language Recognition with Correlation Network*, Version Number: 3, 2023. doi: 10.48550/ARXIV.2303.03202. διεύθν.: https://arxiv.org/abs/2303.03202.

[56] Ethnologue, *Subgroup 2*. διεύθν.: https://www.ethnologue.com/subgroup/2/.

[57] K. Diamantaras και D. Botsis, *Machine Learning*. Kleidarithmos Publications, 2019, isbn: 978-960-461-995-5. διεύθν.: https://www.klidarithmos.gr/mhxanikh-ma8hsh/.

[58] A. Voulodimos, N. Doulamis, A. Doulamis και E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review", en, *Computational Intelligence and Neuroscience*, τόμ. 2018, σσ. 1–13, 2018, issn: 1687-5265, 1687-5273. doi: 10.1155/2018/7068349. διεύθν.: https://www.hindawi.com/journals/cin/2018/7068349/.