National Technical University Of Athens
School of Electrical and Computer Engineering
Division of Signals, Control and Robotics

Deep Learning Based Sign Language Recognition

Diploma Thesis

Maria Parelli

Supervisor: Petros Maragos
Professor NTUA

Computer Vision, Speech Communication and Signal Processing Group

Athens, June 2021

Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Τομεας Σηματων, Ελεγχου και Ρομποτικης

Αναγνώριση Νοηματικής Γλώσσας με Τεχνικές Βαθιάς Μηχανικής Μάθησης

Διπλωματικη Εργασια

Μαρια Παρελλη

Επιβλέπων: Πέτρος Μαραγκός
Καθηγητής Ε.Μ.Π

Εργαστηριο Ορασης Υπολογιστων ,Επικοινωνιας ,Λογου και Επεξεργασιας Σηματων

Αθήνα, Ιούνιος 2021

Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Τομεας Σηματων, Ελεγχου και Ρομποτικης

Αναγνώριση Νοηματικής Γλώσσας με Τεχνικές Βαθιάς Μηχανικής Μάθησης

Διπλωματικη Εργασια

Μαρια Παρελλη

Επιβλέπων: Πέτρος Μαραγκός
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 24η Ιουνίου 2021

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Πέτρος Μαραγκός Καθηγήτης Ε.Μ.Π.	Κωνσταντίνος Τζαφέστας Αναπληρωτής Καθηγητής Ε.Μ.Π	Γεράσιμος Ποταμιάνος Αναπληρωτής Καθηγητής Παν/μιο Θεσσαλίας.

Εργαστηριο Ορασης Υπολογιστων,Επικοινωνιας,Λογου και Επεξεργασιας Σηματων

Αθήνα, Ιούνιος 2021

Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Τομεας Σηματων, Ελεγχου και Ρομποτικης

(Υπογραφή)

.....

Μαρία Παρέλλη

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Abstract

Sign Language constitutes the primary means of communication for the deaf and hard-of-hearing. Sign Language Recognition is a complex task, which lies at the intersection of computer vision and language modeling. Manual and non-manual cues such as expression, hand shape and body orientation occur in parallel and play a meaningful role in the articulation of the sign. In this thesis we study this problem extensively by leveraging recent deep learning approaches.

In the first section we focus on 3D Hand and Body Pose estimation and report quantitative and qualitative results. In the second section we explore the task of continuous sign language recognition and how expressive 3D skeleton and parameterizations of the human body can be exploited in conjunction with graph convolutions in order to effectively solve our task. We also compare our results with successful architectures, such as transformers and LSTM attention encoder-decoders. We report competitive performance on the Phoenix 2014-T dataset.

Περίληψη

Η νοηματική γλώσσα αποτελεί το πρωταρχικό μέσο επικοινωνίας για τους κωφούς και τα άτομα με προβλήματα ακοής. Η αναγνώριση νοηματικής γλώσσας είναι μια πολύπλοκη εργασία, η οποία βρίσκεται στη διασταύρωση της όρασης υπολογιστών και της επεξεργασίας γλώσσας. Χειροκίνητα και μη χειροκίνητα στοιχεία όπως η έκφραση, το σχήμα του χεριού και ο προσανατολισμός του σώματος εξελίσσονται παράλληλα και παίζουν σημαντικό ρόλο στην άρθρωση του νοήματος. Σε αυτή τη διπλωματική εργασία μελετάμε αυτό το πρόβλημα εκτενώς αξιοποιώντας τις πρόσφατες προσεγγίσεις βαθιάς μηχανικής μάθησης.

Στην πρώτη ενότητα εστιάζουμε στην εκτίμηση 3D σκελετού σώματος και χεριού και αναφέρουμε ποσοτικά και ποιοτικά αποτελέσματα. Στη δεύτερη ενότητα διερευνούμε το πρόβλημα της συνεχούς αναγνώρισης νοηματικής γλώσσας και του πώς ο τρισδιάστατος σκελετός και παραμετροποιήσεις του σχήματος του ανθρώπινου σώματος μπορούν να αξιοποιηθούν σε συνδυασμό με συνελίξεις σε γραφήματα προκειμένου να επιλυθεί αποτελεσματικά το έργο μας. Συγκρίνουμε επίσης τα αποτελέσματά μας με επιτυχημένες αρχιτεκτονικές, όπως transformers και αποκωδικοποιητές LSTM με διάφορους μηχανισμούς προσοχής. Αναφέρουμε ανταγωνιστικές επιδόσεις στο σύνολο δεδομένων RWTH Phoenix 2014T.

Ευχαριστίες

Θα ήθελα, κατα πρώτον, να ευχαριστήσω τον Καθηγητή κ. Πέτρο Μαραγκό για την ευκαιρία που μου έδωσε να εκπονήσω τη διπλωματική μου στο εργαστηρίο Ορασης Υπολογιστών, Επικοινωνίας Λόγου και Επεξεργασίας Σημάτων(CVSP). Τα μαθήματα του αποτέλεσαν έμπνευση για εμένα και έπαιξαν καθοριστικό ρόλο στην ανάπτυξη των επιστημονικών μου ενδιαφερόντων. Παράλληλα από την αρχή της συνεργασίας μας μέχρι και σήμερα μου προσέφερε πολύτιμες συμβουλές, με καθοδήγησε σε όλα τα βήματα της έρευνάς μου και συνέβαλε στην εξέλιξή μου ως ερευνήτρια.

Επιπλέον θα ήθελα να ευχαριστήσω προσωπικά τον Καθηγητή κ. Γεράσιμο Ποταμιάνο και την διδακτορική φοιτήτρια Κατερίνα Παπαδημητρίου για την εποικοδομητική συνεργασία μας κατα τη διάρκεια της διπλωματικής μου εργασίας.

Επίσης, ευχαριστώ τους φίλους μου με τους οποίους έχουμε μοιραστεί αξέχαστες εμπειρίες κατα τη διάρκεια της φοίτησης μας. Η συμπαράσταση και την ενθάρρυνση τους ήταν πολύτιμη και χωρίς αυτούς τα χρόνια φοίτησής μου δεν θα ήταν το ίδιο όμορφα.

Τέλος, δεν μπορώ παρα να ευχαριστήσω του γονείς μου για την υποστήριξη και κατανόησή που μου έδειξαν καθόλη τη διάρκεια των σπουδών μου.

Acknowledgements

First of all, I would like to thank Professor Petros Maragos for the opportunity to conduct my diploma thesis in the Computer Vision, Communication and Signal Processing Laboratory(CVSP). His lessons were an inspiration to me and played a key role in the development of my scientific interests. At the same time, from the beginning of our collaboration until today, he offered me valuable advice, guided me through all the steps of my research and contributed to my development as a researcher.

In addition, I would like to personally thank Professor Gerasimos Potamianos and PhD student Katerina Papadimitriou for our constructive cooperation during my thesis.

I also thank my friends with whom we have shared unforgettable experiences during our studies. Their support and encouragement was invaluable and without them my years of study would not have been as memorable.

Finally, I would never forget to thank my parents for the support and understanding they have shown me throughout my studies.

Contents

List of Figures

List of Tables

Κεφάλαιο 1

Εκτενής Περίληψη

1.0.1 Εισαγωγή

Οι άνθρωποι τις τελευταίες δεκαετίες επιδιώκουν να παρέχουν στις μηχανές τη δυνατότητα να «σκέφτονται» και να διαμορφώνουν μια κατανόηση των τρισδιάστατων περιβαλλόντων τους και των ενεργειών και των εκφράσεων των ανθρώπων. Οι μέθοδοι αναγνώρισης χειρονομίας και δράσης που βασίζονται σε όραση υπολογιστών στοχεύουν στη διευκόλυνση της αποτελεσματικής και φυσικής επικοινωνίας και αλληλεπίδρασης μεταξύ ανθρώπου-μηχανής.

Τα προηγούμενα χρόνια γάντια δεδομένων και δείκτες και αργότερα παραδοσιακές μεθοδολογίες όρασης υπολογιστών, όπως οι περιγραφητές χαρακτηριστικών και οι μηχανές διανυσμάτων υποστήριξης SVMs κυριάρχησαν στο πεδίο. Η έλευση της βαθιάς μηχανικής μάθησης έφερε επανάσταση στο τοπίο και επέτρεψε νέες ισχυρές προσεγγίσεις σε αυτόν τον τομέα που βασίζονται στη διαθεσιμότητα συνόλων δεδομένων μεγάλης κλίμακας. Σε αυτή τη διατριβή αντιμετωπίζουμε το πρόβλημα της αναγνώρισης νοηματικής γλώσσας, το οποίο μπορούμε να θεωρήσουμε ότι συνδυάζει το πρόβλημα της αναγνώρισης χειρονομιών και της εκτίμησης της πόζας του σώματος.

1.0.1.1 Ορισμός προβλήματος

Η νοηματική γλώσσα είναι η γλώσσα των Κωφών και των ατόμων με προβλήματα ακοής και αποτελεί το κύριο μέσο επικοινωνίας τους. Η αναγνώριση νοηματικής γλώσσας είναι ένα δύσκολο και πολύπλοκο έργο όρασης υπολογιστών που συνδυάζει όραση υπολογιστών και μοντελοποίηση φυσικής γλώσσας. Πολλαπλές ενδείξεις, όπως η έκφραση του προσώπου, το σχήμα και η πόζα του χεριού και οι κινήσεις του σώματος συνδυάζονται για την αποτελεσματική μεταφορά πληροφοριών. Τα μοντέλα νοηματικής γλώσσας πρέπει να εξαγάγουν μια πλούσια χωροχρονική αναπαράσταση από ένα βίντεο εισόδου και να κατανοήσουν πώς κινείται ένας νοηματιστής μέσα σε έναν τρισδιάστατο χώρο. Υπό αυτήν την πτυχή, η αναγνώριση

νοηματικής γλώσσας μπορεί να θεωρηθεί υπο-πρόβλημα της αναγνώρισης δράσεων και χειρονομιών. Ωστόσο, ο στόχος ενός συστήματος αναγνώρισης είναι να προβλέψει την ακολουθία των νοημάτων, που εκτελείται από τον νοηματιστή. Η νοηματική είναι μια ολοκληρωμένη γλώσσα, με όλες τις ιδιαιτερότητες και τις λεπτές αποχρώσεις μίας ομιλούμενης γλώσσας, γεγονός που καθιστά το αναγνώρισή της ένα σύνθετο έργο γλωσσικής μοντελοποίησης. Η έρευνα στην όραση υπολογιστών είχε επικεντρωθεί εκτενώς στην απομονωμένη αναγνώριση νοηματικής γλώσσας. Με την εμφάνιση βάσεων δεδομένων αναγνώρισης μεγάλης κλίμακας, η έρευνα έχει μετατοπιστεί στη συνεχή αναγνώριση νοηματικής γλώσσας.

Η τρισδιάστατη εκτίμηση της πόζας του χεριού είναι μια υπο-περιοχή της Όρασης Υπολογιστών που στοχεύει στην εξαγωγή τρισδιάστατης αναπαράστασης των αρθρώσεων του ανθρώπινου χεριού. Η τοποθεσία, ο προσανατολισμός και η άρθρωση του χεριού στον τρισδιάστατο χώρο είναι χρήσιμα στοιχεία για μια πληθώρα εφαρμογών, συμπεριλαμβανομένης της αναγνώρισης νοηματικής γλώσσας στην οποία οι λεπτές κινήσεις των δακτύλων παίζουν σημαντικό ρόλο. Πολλοί ερευνητές στο παρελθόν χρησιμοποιούσαν πολλαπλές κάμερες και πληροφορίες βάθους για να εξάγουν τρισδιάστατη πόζα, αλλά σε αυτή τη διατριβή επικεντρωνόμαστε σε νέες μεθόδους που χρησιμοποιούν εικόνες RGB και μπορούν να γενικεύουν αποτελεσματικά σε παραδείγματα στα οποία δεν έχουν εκπαιδευτεί.

1.0.1.2 Συνεισφορές

Η συμβολή μας σε αυτή τη διατριβή κινείται σε δύο κατευθύνσεις. Πρώτον, αναγνωρίζοντας τη σημασία της γνώσης της λεπτομερούς πορείας των χεριών και των δακτύλων, προτείνουμε μεθόδους που είναι επιτυχείς στην εξαγωγή των τρισδιάστατων αρθρώσεων σώματος και χεριών των νοηματιστών και μπορούν να γενικευτούν σε βίντεο στα οποία δεν έχουν εκπαιδευτεί.

Δεύτερον, αφού αποκτήσουμε μια πιο πλούσια αναπαράσταση του ανθρώπινου σκελετού, διερευνούμε πώς μπορεί να αξιοποιηθεί το πλήρες σχήμα και η πόζα του σώματος και να συγχωνευθούν αποτελεσματικά με οπτικά χαρακτηριστικά με στόχο την αναγνώριση συνεχούς νοηματικής γλώσσας. Πραγματοποιούμε τα πειράματά μας και παρατηρούμε ανταγωνιστική απόδοση σε σύγκριση με τις τρέχουσες στο σύνολο δεδομένων RWTH Phoenix 2014T.

- Εμπνευσμένοι από επιτυχείς προσεγγίσεις σε εκτίμηση τρισδιάστατης πόζας χεριού και σώματος, εφαρμόζουμε μια βαθιά πλήρως συνδεδεμένη αρχιτεκτονική με γεωμετρικούς περιορισμούς για τρισδιάστατη εκτίμηση της πόζας χεριού και δοκιμάζουμε διεξοδικά την απόδοσή της σε δύο σύνολα δεδομένων με 3D χειρομορφές. Μετράμε επίσης τη δύναμη γενίκευσης της μεθόδου μας και παρέχουμε ποιοτικά παραδείγματα των προβλέψεών της όταν εικόνες νοηματικής γλώσσας χρησιμοποιούνται ως είσοδοι.

- Η διπλωματική αυτή αποτελεί ένα από τα πρώτα έργα που συνδυάζουν 3D πλήρες σχήμα σώματος και παραμέτρους πόζας και εκμεταλλεύονται μια πλούσια τρισδιάστατη αναπαράσταση του ανθρώπινου σώματος στο πρόβλημα της αναγνώρισης της νοηματικής γλώσσας.

- Αξιοποιούμε τη δύναμη των γραφοσυνελικτικών δικτύων, τα οποία δεν έχουν διερευνηθεί διεξοδικά στη συνεχή αναγνώριση νοηματικής γλώσσας. Η τρισδιάστατη θέση και την αναπαράσταση περιστροφής άξονα-γωνίας της άρθρωσης ενσωματώνεται στους κόμβους γραφήματος και συνδυάζεται με οπτικά χαρακτηριστικά. Επίσης, προσθέτουμε έναν κωδικοποιητή LSTM μετά τα στρώματα συνέλιξης για χρονική μοντελοποίηση, ο οποίος παρουσιάζει αυξημένη απόδοση.

- Εξετάζεται τη συνεισφορά μιας τεχνικής σταθμισμένης σύντηξης πιθανοτήτων, όπου οι posteriori πιθανότητες των διαφορετικών μοντέλων συγχρονίζονται και συγχωνεύονται για την πρόβλεψη της ακολουθίας νοημάτων.

- Αναφέρουμε ανταγωνιστικά αποτελέσματα, ξεπερνώντας πολλές σύγχρονες προσεγγίσεις στο απαιτητικό σύνολο δεδομένων RWTH Phoenix 2014T και συγκρίνουμε τη μέθοδο μας με ισχυρές αρχικές αρχιτεκτονικές μετασχηματιστών νοηματικής γλώσσας και αποκωδικοποιητές με διαφορετικούς μηχανισμούς προσοχής.

1.1 Τρισδιάστατη Εκτίμηση Σκελετού Χεριού και Σώματος

Η άρθρωση του χεριού παίζει πρωταρχικό ρόλο στην αναγνώριση νοηματικής γλώσσας και πολλά νοήματα χαρακτηρίζονται από παρόμοια μοτίβα κίνησης του σκελετού. Συνεπώς, είναι ζωτικής σημασίας να αναζητηθούν μέθοδοι που εμπλουτίζουν την κίνηση των χεριών και τις δομικές πληροφορίες και παρέχουν στα μοντέλα σημαντικές γνώσεις σχετικά με την πορεία των χεριών και των δακτύλων στο τρισδιάστατο επίπεδο. Στο πρώτο μέρος της διατριβής μας στρέψαμε την προσοχή μας σε μεθόδους εκτίμησης τρισδιάστατης θέσης, οι οποίες μπορούν να ανυψώσουν αποτελεσματικά τις συντεταγμένες 2D στον 3D χώρο.

Η προσέγγισή μας εξάγει τρισδιάστατες θέσεις των άρθρωσεων του χεριού «ανυψώνοντας» 2D θέσεις αρθρώσεων στον 3D χώρο. Η είσοδος μας είναι μια σειρά από 2D σημεία, που είχαν εξαχθεί προηγουμένως από το δίκτυο OpenPose [6] και η έξοδος μας είναι μια σειρά σημείων στον χώρο 3D. Εμείς θέτουμε ως αρχή των αξόνων την άρθρωση του καρπού, ώστε να διασφαλίσουμε ότι το μοντέλο μας μαθαίνει χωρικά αναλλοίωτες αναπαραστάσεις. Μια σημαντική πηγή σφάλματος στις τρισδιάστατες προβλέψεις είναι η ύπαρξη θορύβου στις

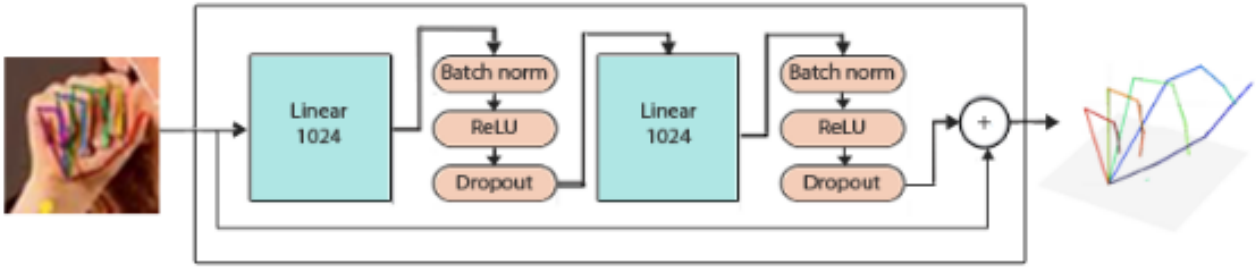
προβλέψεις εισόδου 2D. Έτσι, εφαρμόζουμε εξομάλυνση στην είσοδο μέσω ενός median φίλτρου με ακτίνα 1 για να αφαιρέσουμε τις αιχμές θορύβου και να εξαλείψουμε την αστάθεια στις προβλέψεις.

$$\tilde{\alpha}[n] = m(\alpha[n-T], ..., \alpha[n], ...\alpha[n+T])$$

, όπου το $\tilde{a}$ δηλώνει το προεπεξεργασμένο σήμα, το $m()$ είναι ο μέσος τελεστής και το 2T + 1 είναι το μέγεθος του φίλτρου.

Στη συνέχεια, εφαρμόζουμε μια απλή αλλά ισχυρή αρχιτεκτονική, που προτάθηκε αρχικά στο [36] για εκτίμηση 3D σκελετού σώματος. Το μοντέλο μας είναι ένα βαθύ πλήρως συνδεδεμένο νευρωνικό δίκτυο πολλαπλών επιπέδων με batch normalization, dropout, ενεργοποίηση (ReLU) και υπολειμματικές συνδέσεις (residual connections). Το τελευταίο ενισχύει την ικανότητα γενίκευσης, ενώ το batch normalization και dropout βελτιώνουν την αντοχή του μοντέλου σε θορυβώδεις 2D ανιχνεύσεις. Επιπλέον, εφαρμόζεται ένας περιορισμός στα βάρη κάθε στρώματος, έτσι ώστε η απόλυτη τιμή τους να είναι μικρότερη ή ίση της μονάδας.

Συγκεκριμένα, το βασικό δομικό στοιχείο του δικτύου είναι ένα γραμμικό πλήρως συνδεδεμένο επίπεδο που ακολουθείται από batch normalization [20], dropout [52] και ενεργοποίηση ReLU. Αυτό το μπλοκ επαναλαμβάνεται δύο φορές και τα δύο μπλοκ μοιράζονται μια υπολειμματική σύνδεση. Για τους σκοπούς της εργασίας αυτής συνενώνουμε δύο μπλοκ και το μοντέλο μας περιέχει περίπου 4 εκατομμύρια παραμέτρους.



Σχημα 1.1: Τα βασικά δομικά στοιχεία του μοντέλου εξαγωγής τρισδιάστατης πόζας χεριού.

Για εκπαίδευση του δίκτυου, χρησιμοποιούμε το Rendered HandPose Dataset [61], ένα μεγάλο σύνολο δεδομένων με 3D πόζες χεριών βασισμένο σε συνθετικά μοντέλα χεριών. Το μοντέλο αποδίδει 21 τρισδιάστατες θέσεις συνδέσμων για κάθε χέρι. Συνεπώς εξάγονται διανύσματα χαρακτηριστικών με διάσταση 126. Ο καρπός θεωρείται ως αρχή του συστήματος συντεταγμένων.

1.1.0.1 Πειραματική αξιολόγηση

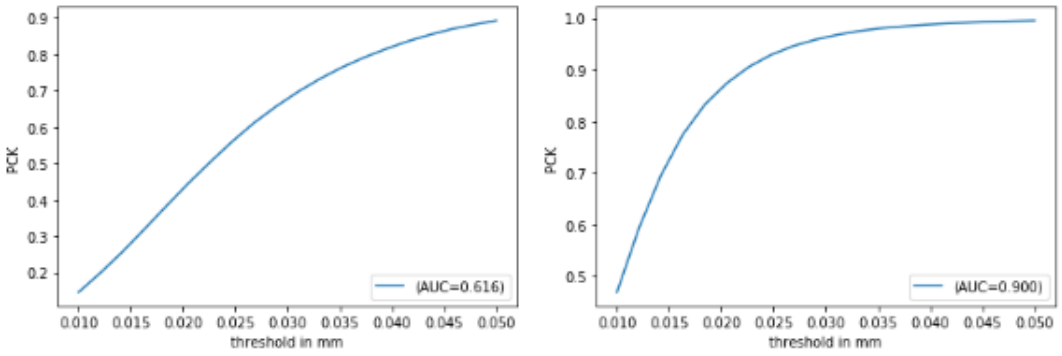
Αξιολογούμε την απόδοση του δικτύου μας χρησιμοποιώντας δύο σύνολα δεδομένων: το σύνολο δεδομένων Rendered HandPose (RHD) και τη βάση δεδομένων FreiHand (FHD) [62].

Στον παρακάτω πίνακα και σχήμα, αξιολογούμε την απόδοση και τη δύναμη γενίκευσης του μοντέλου μας στα προαναφερθέντα σύνολα δεδομένων για διάφορες ρυθμίσεις εκπαίδευσης / δοκιμών. Αναφέρουμε το μέσο σφάλμα ανά άρθρωση της προβλεπόμενης τρισδιάστατης πόζας, όταν δίνεται ο 2D σκελετός, καθώς και τη περιοχή κάτω από την καμπύλη (AUC) στο ποσοστό των σωστών προβλέψεων θέσης για διαφορετικά όρια σφάλματος.

Συγκεκριμένα, για να διερευνήσουμε τη γενίκευση του δικτύου μας, χρησιμοποιούμε το μοντέλο που εκπαιδεύτηκε σε RHD-train και αναφέρουμε τη βαθμολογία AUC και το μέσο σφάλμα ανά άρθρωση στο σύνολο δεδομένων FHD, μετά την ευθυγράμμιση με την πραγματική θέση στο χώρο (ανάλυση Procrustes). Αναφέρουμε επίσης το ποσοστό των σωστά προβλεπόμενων σημείων (PCK), το οποίο επιστρέφει το μέσο ποσοστό των προβλεπόμενων αρθρώσεων που βρίσκονται εντός μιας ευκλείδειας απόστασης από τη σωστή θέση της άρθρωσης. Τα αποτελέσματα φανερώνουν ότι η μέθοδος μας δείχνει πολύ καλή απόδοση και στα δύο σύνολα δεδομένων.

Το σετ RHD μπορεί να χαρακτηριστεί ως απαιτητικό, λόγω των διαφορών σε οπτικές γωνίες της κάμερας, και ως εκ τούτου αναφέρουμε υψηλότερο σφάλμα 3D σκελετού. Δεδομένου ότι ενδιαφερόμαστε ως επί το πλείστον για τη δύναμη γενίκευσης του μοντέλου μας και την απόδοσή του σε παραδείγματα στα οποία δεν έχει εκπαιδευτεί, παρατηρούμε ότι το μοντέλο μας καταφέρνει να προσαρμοστεί αποτελεσματικά σε νέα δεδομένα και συλλαμβάνει με ακρίβεια τον σκελετό του χεριού.

Training	Testing	AUC score	Median error per joint
RHD-Train	RHD-train	0.729	18.1
	RHD-test	0.616	22.6
	FHD-test	0.771	16.2
FHD-Train	FHD-test	0.900	11.0



Σχημα 1.2: Ποσοστό σωστών σημείων (PCK) κάτω από ένα ορισμένο κατώφλι σε mm, που αξιολογήθηκε: α) στο RHD-test για μοντέλο εκπαιδευμένο στο RHD-train και (β) στο RHD-test για μοντέλο εκπαιδευμένο στο FHD-train.

Οι μικρές απώλειες στον τρισδιάστατο ευκλείδειο χώρο δεν μεταφράζονται πάντα σε αμελητέες παραμορφώσεις. Βασιζόμενοι στη μέθοδο, που εισήχθη στο [17], η οποία εκτιμά την τρισδιάστατη πόζα χεριών μέσω βελτιστοποίησης μοντέλου χεριών που βασίζεται σε γραφήματα κάτω από ένα πλαίσιο ανταγωνιστικής μάθησης, χρησιμοποιήσαμε δύο επιπλέον συναρτήσεις απώλειών και τις ενσωματώσαμε στο μοντέλο μας: μια συνάρτηση μήκους οστού L_{len}, η οποία υπολογίζει τη διαφορά μεταξύ του πραγματικού μήκους των οστών και της προβλεψης τους και μια συνάρτηση κατεύθυνσης των οστών L_{dir}, η οποία μετρά την απόκλιση στην κατεύθυνση των οστών. Συγκεκριμένα,

$$L_{len} = \sum_{i=1}^{N}\sum_{j=1}^{N}\left|\left\|b_{ij}\right\| - \left\|\hat{b_{ij}}\right\|\right|$$

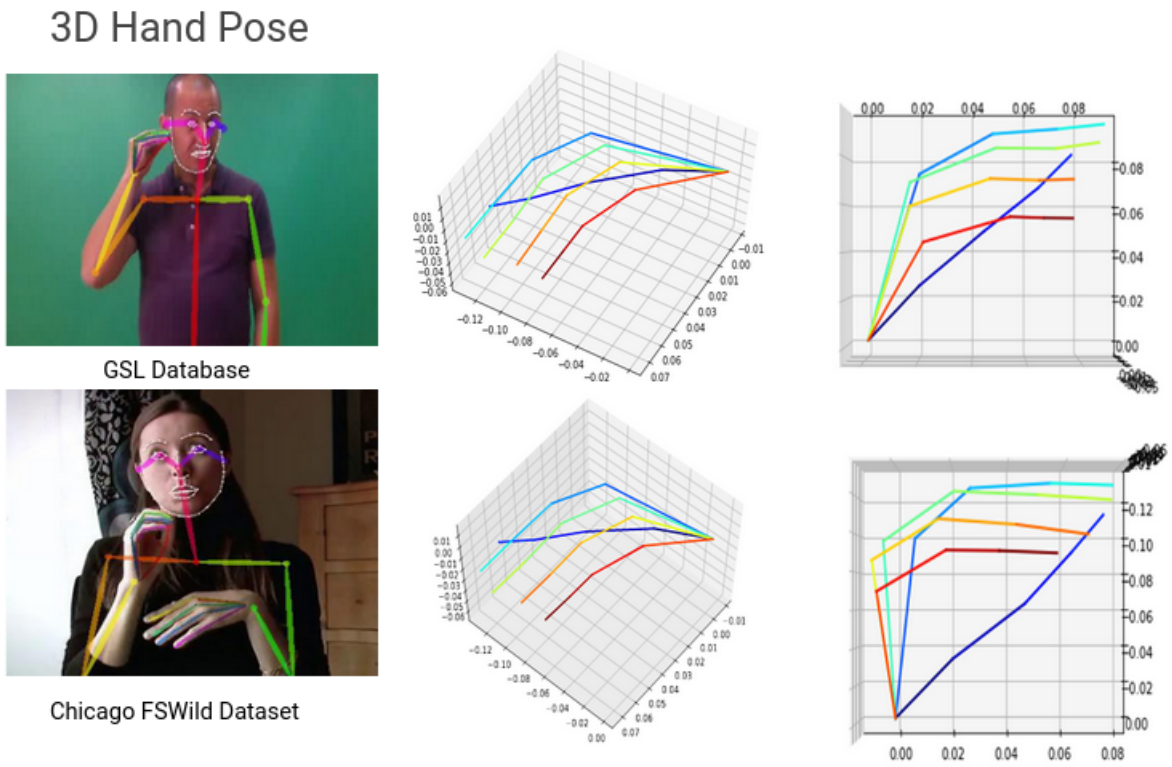
$$L_{dir} = \sum_{i=1}^{N}\sum_{j=1}^{N}\left\|\frac{b_{ij}}{\left\|b_{ij}\right\|} - \frac{\hat{b_{ij}}}{\left\|\hat{b_{ij}}\right\|}\right\|$$

όπου ο όρος $b_{ij} = j_i - j_j$ υποδηλώνει το διάνυσμα του οστού μεταξύ των αρθρώσεων i και j Η τελική συνάρτηση απωλειών είναι ο σταθμισμένος μέσος όρος:

$$L_{loss} = L_{joint} + \lambda_{len}L_{len} + \lambda_{dir}L_{dir}$$

όπου ο όρος L_{joint} αναπαριστά την 3D ευκλίδεια απόσταση μεταξύ της πραγματικής άρθρωσης και της εκτίμησης του μοντέλου.

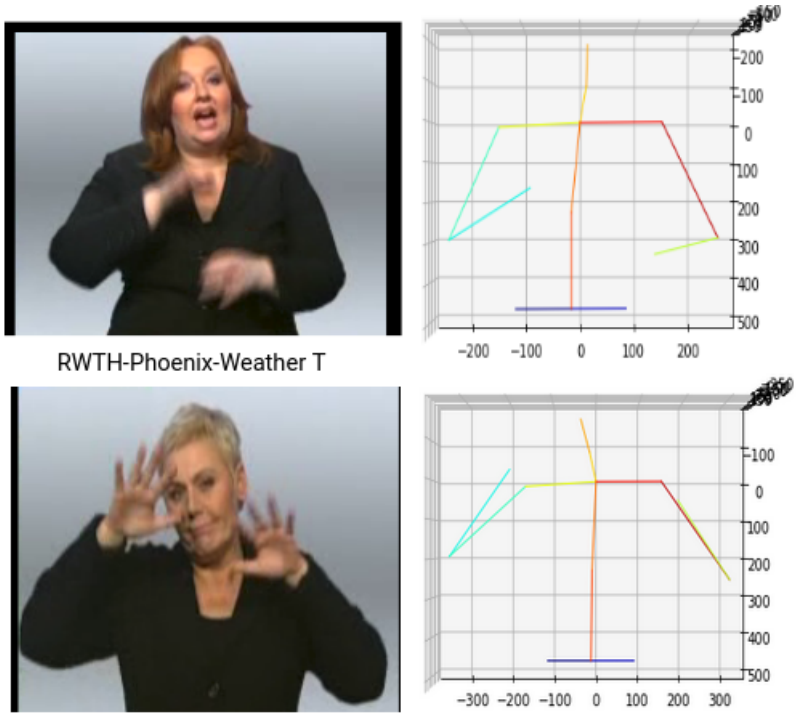
Επίσης, δοκιμάζουμε ποιοτικά τη μέθοδο μας, προκειμένου να προσδιορίσουμε αν μπορεί να γενικεύσει σε βίντεο "in the wild". Τα ακόλουθα παραδείγματα είναι τρισδιάστατοι σκελετοί χεριών που παράγονται από το μοντέλο μας με είσοδο βίντεο νοηματικής γλώσσας.



Σχημα 1.3: Παραδείγματα εκτίμησης τρισδιάστατης πόζας χεριού στο σύνολο δεδομένων GSLL [54] και στο Chicago FSWild Dataset.

Η μορφή των χεριών παίζει καθοριστικό ρόλο στην αναγνώριση της νοηματικής γλώσσας. Ωστόσο, η γνώση σχετικά με την τροχιά των αρθρώσεων των σώματος στο τρισδιάστατο επίπεδο μπορεί επίσης να δώσει σημαντικές πληροφορίες και να βοηθήσει το μοντέλο να διακρίνει μεταξύ νοημάτων που παρουσιάζουν παρόμοια χειρομορφή

Η ίδια αρχιτεκτονική μπορεί να χρησιμοποιηθεί για την ανύψωση των σημείων του σώματος στον 3D χώρο, όπως προτείνεται στο πρωτότυπο έργο. Εκπαιδεύουμε εκ νέου το δίκτυο για 200 εποχές, χρησιμοποιώντας ένα υποσύνολο του συνόλου δεδομένων Human 3.6 M [9, 21]. Δεδομένου ότι ο κύριος στόχος είναι η αναγνώριση νοηματικής γλώσσας και σε αυτήν τη ρύθμιση συνήθως είναι ορατό μόνο το άνω μέρος του σώματος, θέτουμε τα σημεία εκτός πλαισίου στο 0 κατά τη διάρκεια της εκπαίδευσης, έτσι ώστε το μοντέλο να μπορεί να προσαρμοστεί στις πόζες, όπου λείπει το κάτω μέρος του σώματος. Αναφέρουμε ένα μέσο σφάλμα 20.3 ανά άρθρωση.



Σχημα 1.4: Παραδείγματα εκτίμησης τρισδιάστατης πόζας σώματος στο σύνολο δεδομένων Phoenix 2014T

1.2 Αναγνώριση Νοηματικής Γλώσσας

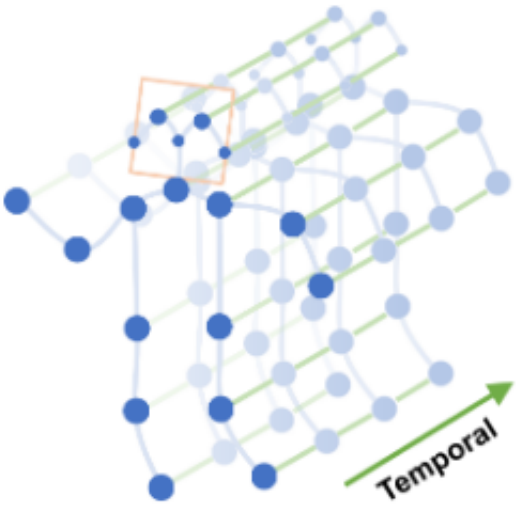
1.2.1 Χωρικά-χρονικά γραφοσυνελικτικά δίκτυα

Η μέθοδος μας εμπνέεται από το [50] που εφαρμόζει GCNs στο έργο της αναγνώρισης δράσης βάσει σκελετού και προτείνει ένα χωρικό-χρονικό γραφοσυνελικτικό δίκτυο (ST-GCN), το οποίο εξάγει αυτόματα τόσο τα χωρικά όσο και τα χρονικά μοτίβα από δεδομένα. Η συνδεσιμότητα των αρθρώσεων αποτελεί μια σημαντική πτυχή του προβλήματος και είναι χρήσιμο να αποτυπωθούν οι συσχετίσεις μεταξύ τους.

Το προτεινόμενο μοντέλο διαμορφώνεται με βάσει μια ακολουθία γραφημάτων σκελετού, όπου κάθε κόμβος αντιστοιχεί σε άρθρωση και υπάρχουν δύο τύποι ακμών, οι χωρικές ακμές που ακολουθούν τη φυσική συνδεσιμότητα των αρθρώσεων και οι χρονικές ακμές, που συνδέουν τις ίδιες αρθρώσεις σε διαδοχικά χρονικά βήματα.

Συγκεκριμένα, κατασκευάζεται ένα μη κατευθυνόμενο γράφημα $G = (V, E)$ όπου ο κόμβος ορίζεται ως $V = (u_{ti}|t = 1, ..., T, i = 1, ..., N)$. Το σύνολο των ακμών αποτελείται από δύο υποσύνολα, το πρώτο υποσύνολο απεικονίζει τη συνδεσιμότητα του σκελετού σε κάθε καρέ, που υποδηλώνεται ως $E_s = (u_{ti}, u_{tj}|(i, j) \in H)$ όπου H είναι το σύνολο των συνδεδεμένων αρθρώσεων του ανθρώπινου σώματος. Το δεύτερο υποσύνολο περιέχει τις ακμές μεταξύ

διαδοχικών στιγμιοτύπων του βίντεο, που υποδηλώνονται ως $E_f = (u_{ti}u_{t+1,i})$. Οι ακμές που ανήκουν στο E_f για μια συγκεκριμένη άρθρωση αντιπροσωπεύουν την τροχια της στην πάροδο του χρόνου.



Σχημα 1.5: Χωροχρονική γραφική αναπαράσταση της ακολουθίας σκελετού [50].

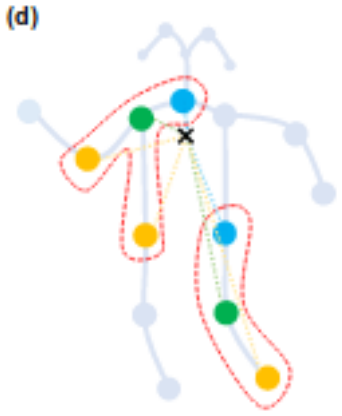
Πρώτον, παρουσιάζουμε τη λειτουργία χωρικής συνέλιξης σε γραφήματα, η οποία ορίζεται επεκτείνοντας τον ορισμό της συνέλιξης σε 2D πλέγματα σε περιπτώσεις όπου ο χάρτης χαρακτηριστικών βρίσκεται σε γράφημα V_t. Στα γραφήματα ορίζουμε το γειτονικό σύνολο ενός κόμβου ως $B(u_{ti}) = u_{tj}|d(u_{tj}, u_{ti}) \leq D$. Σε αυτήν την εργασία χρησιμοποιείται $D = 1$, δηλαδή το γειτονικό σύνολο κόμβων σε απόσταση 1.

Για να καθορίσουμε τη συνάρτηση βάρους, χωρίζουμε το γειτονικό σετ $B_{u_{ti}}$ σε έναν σταθερό αριθμό υποομάδων, όπου κάθε υποσύνολο αντιστοιχεί σε μια ετικέτα. Έτσι έχουμε μια χαρτογράφηση l_{ti}, η οποία χαρτογραφεί έναν κόμβο στην ετικέτα υποσυνόλου του. Η συνάρτηση βάρους μπορεί να εφαρμοστεί μέσω indexing σε (c, K) διάσταση όπου K είναι ο αριθμός των υποομάδων.

Είναι σημαντικό να σχεδιαστεί μια αποτελεσματική στρατηγική διαμέρισης. Εδώ χρησιμοποιείται στρατηγική χωρικής διαμέρισης (spatial partitioning). Το γειτονικό σύνολο χωρίζεται σε τρία υποσύνολα: 1) τον ριζικό κόμβο. 2) την κεντρομόλο ομάδα: οι γειτονικοί κόμβοι που είναι πιο κοντά στο κέντρο βαρύτητας του σκελετού από τον ριζικό κόμβο. 3) διαφορετικά την φυγοκεντρική ομάδα.

Συγκεκριμένα:

$$l_{ti}(u_{tj}) = \begin{cases} 0 & if\ r_j = r_i \\ 1 & if\ r_j < r_i \\ 2 & if\ r_j > r_i \end{cases}$$



Σχημα 1.6: Στρατηγική χωρικής διαμέρισης.

1.2.1.1 Χωροχρονική συνέλιξη σε γραφήματα

Η χωροχρονική συνέλιξη σε γραφήματα ορίζεται ως εξής:

$$f_{out}(u_{ti}) = \sum_{u_{tj} \in B_{u_{t_ij}}} \frac{1}{Z_{ti}(u_{tj})} f_{in}(u_{tj}) \cdot w(l_{ti}(u_{tj})$$

Μετά τη διαμόρφωση της συνέλιξης σε γραφημάτα, το επόμενο βήμα είναι να επεκταθεί το χωρικό γράφημα στον χωροχρονικό τομέα. Έτσι, συμπεριλαμβάνουμε χρονικά συνδεδεμένους συνδέσμους στη γειτονιά ως :

$$B_{u_{ti}} = u_{qj} | d(u_{tj}, u_{ti}) \leq K, |q - t| \leq \lfloor \Gamma/2 \rfloor$$

όπου η παράμετρος Γ ελέγχει τη χρονική γειτονιά του κόμβου. Ο χάρτης ετικετών l_{ST} τροποποιείται ως εξής:

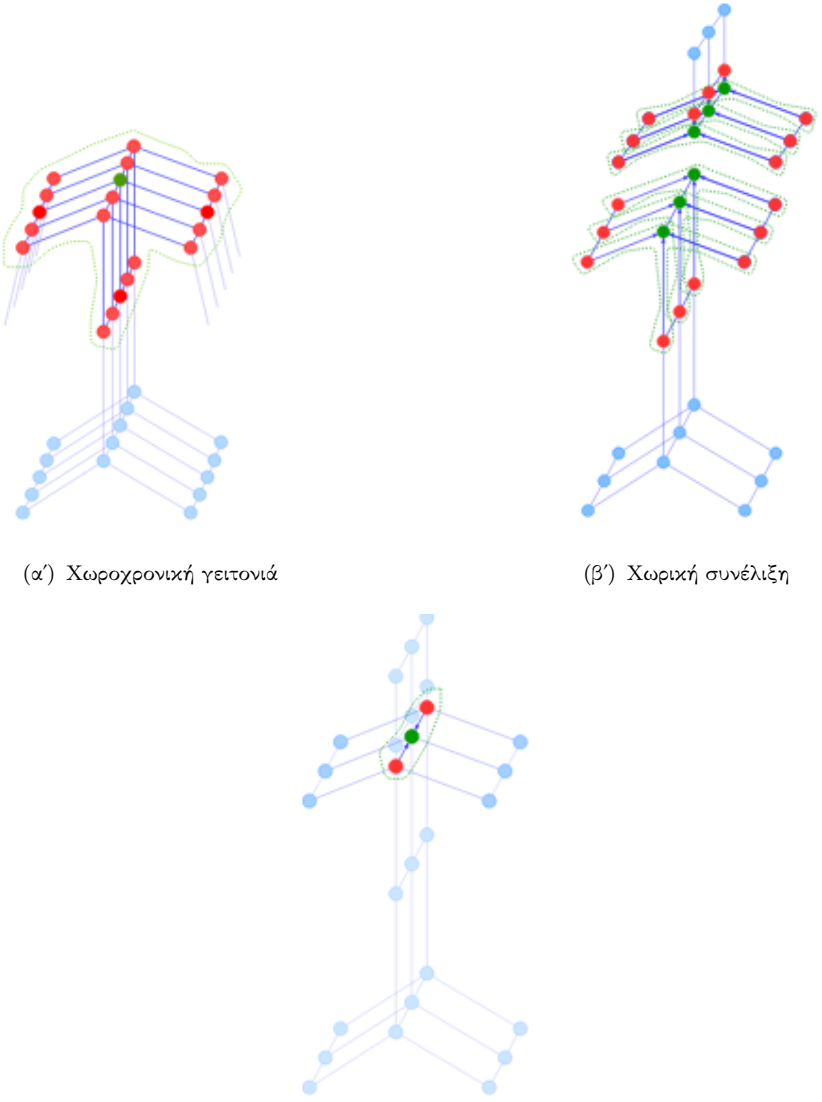
$$l_{ST}(u_{qj}) = l_{ti}(u_{tj}) + (q - t + \lfloor \Gamma/2 \rfloor) \times K$$

Υλοποίηση ST-GCN

Για μια στρατηγική διαμέρισης με πολλαπλά υποσύνολα, ο πίνακας γειτνίασης A αποτελείται από πίνακες A_j όπου $A + I = \sum_j A_j$. Έτσι, το ST-GCN στην περίπτωση ενός πλαισίου μπορεί να οριστεί ως:

$$f_{out} = \Lambda_j^{-1/2} A_j \Lambda_j^{-1/2} f_{in} W_j$$

όπου Λ είναι ο διαγώνιος πίνακας βαθμών και W είναι ο πίνακας βαρών



(α′) Χωροχρονική γειτονιά

(β′) Χωρική συνέλιξη

(γ′) Χρονική γειτονιά

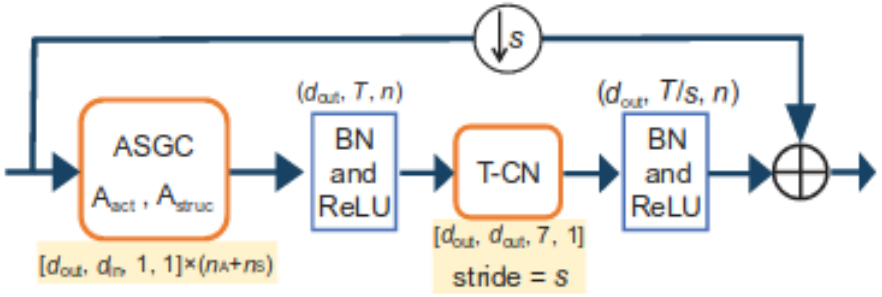
Σχημα 1.7: Ορισμός γειτονιών [50]

Ωστόσο, η εμφάνιση μιας άρθρωσης σε πολλά μέρη του σώματος θα πρέπει να έχει διαφορετική σημασία στη μοντελοποίηση της δυναμικής αυτών των μερών. Επομένως, είναι φυσικό να προστεθεί μια μάσκα M σε κάθε στρώμα της χωρικής χρονικής συνέλιξης γραφήματος, η οποία θα κλιμακώσει τη συνεισφορά ενός κόμβου στους γειτονικούς κόμβους του με βάση το βάρος κάθε ακμής χωρικού γραφήματος σε E_s. Στην προηγούμενη εξίσωση η A_j αντικαθίσταται με $A_j \otimes M$. Η μάσκα M αρχικοποιείται ως μοναδιαίος πίνακας .

Αρχιτεκτονική πλαισίου ST-GCN

Το δίκτυο κορμού αποτελείται από μια σειρά ST-GCN πλαισίων. Τα στρώματα της γραφικής συνέλιξης και των χρονικών συνελίξεων συνενώνονται για να σχηματίσουν ένα μπλοκ ST-GCN που εξάγει τόσο χωρικές όσο και χρονικές πληροφορίες. Το μπλοκ ST-GCN

αποτελείται επίσης από άλλες λειτουργίες: batch normalization, ReLU ενεργοποίηση και υπολειμματική συνδεση (residual connection).

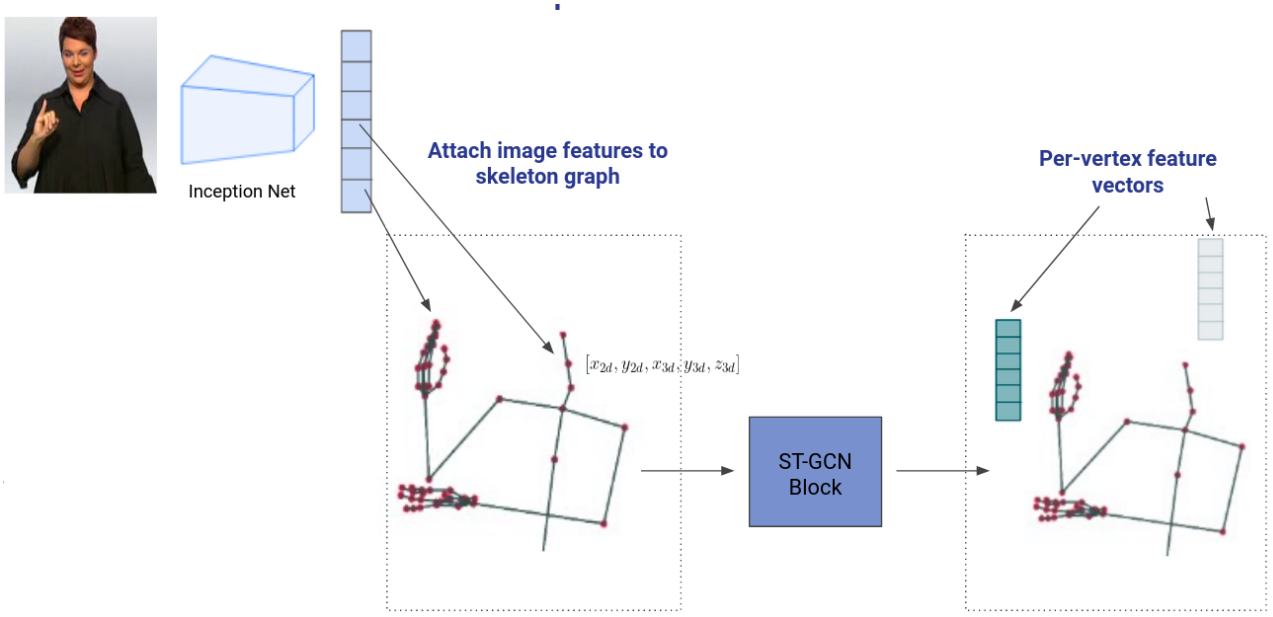


Σχημα 1.8: Δομή μονάδας ST-GCN [50] .

1.2.1.2 Αναγνώριση νοηματικής γλώσσας με τρισδιάστατη πληροφορία και συνελιξη σε γραφημάτα

Στα πειράματά μας εστιάζουμε στον τρόπο με τον οποίο ο 3D σκελετός και το σχήμα μπορούν να ενσωματωθούν αποτελεσματικά στον μοντέλο αναγνώρισης, ώστε να εμπλουτίσουν τις γνώσεις του μοντέλου σχετικά με το νόημα. Χρησιμοποιώντας το μπλοκ ST-GCN ως κύρια μονάδα δόμησης, χτίζουμε ένα δίκτυο που συνδυάζει τις παραμέτρους πόζας και σχήματος με οπτικά χαρακτηριστικά [30]. Πιο συγκεκριμένα, η αρχιτεκτονική Graph CNN μας επιτρέπει να κωδικοποιήσουμε τη σκελετική δομή μέσα στο δίκτυο και να αξιοποιήσουμε τη χωρική τοποθεσία των αρθρώσεων. Με δεδομένο ένα πλαίσιο εισόδου, ένας κωδικοποιητής CNN εξάγει χαρακτηριστικά εικόνας από την αναπαράσταση εισόδου RGB. Στη συνέχεια, αυτά τα χαρακτηριστικά συνενώνονται με τις 2D και 3D συντεταγμένες άρθρωσεων χεριού και σώματος.

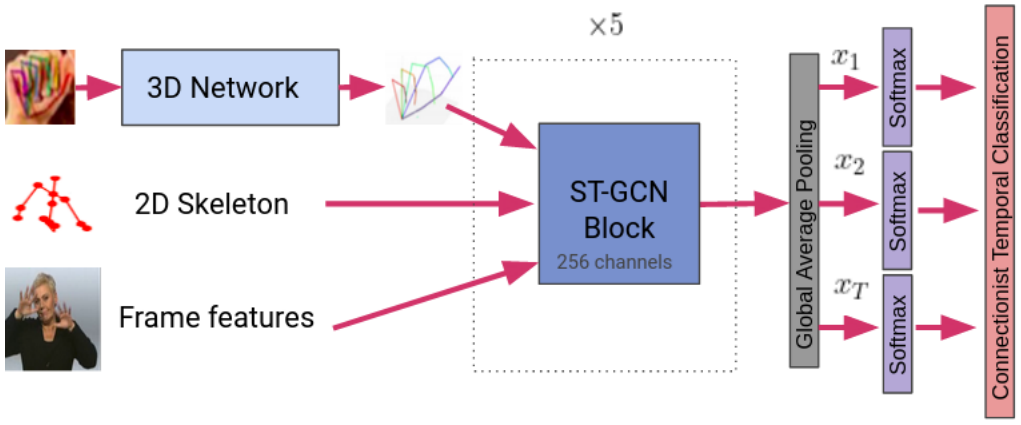
Το προσαρμοσμένο δίκτυο ST-GCN χρησιμοποιεί ως είσοδο τις τρισδιάστατες συντεταγμένες κάθε άρθρωσης μαζί με τα οπτικά χαρακτηριστικά της εισόδου και σε κάθε επίπεδο συγκεντρώνει πληροφορίες από τη χωρική-χρονική γειτονιά κάθε κόμβου για να αποκτήσει μια αναπαράσταση χαρακτηριστικών ανά κορυφή.



Σχημα 1.9: Αναπαράσταση της προτεινόμενης μεθόδου.

Αρχιτεκτονική Μοντέλου

Η αρχιτεκτονική μας αποτελείται από πέντε επίπεδα ST-GCN μπλοκ με 256 κανάλια το καθένα, ακολουθούμενο από ένα επίπεδο συγκέντρωσης (global pooling) ανά κόμβο. Η έξοδος είναι ένα διάνυσμα 256 χαρακτηριστικών για κάθε στιγμιότυπο βίντεο και τροφοδοτείται σε ένα γραμμικό softmax επίπεδο για την τελική πρόβλεψη. Η εξαγωγή πρόβλεψης πραγματοποιείται μέσω ενός αποκωδικοποιητή δεσμικής αναζήτησης CTC (beam search) με μέγεθος δέσμης $= 3$. Για εξαγωγή χαρακτηριστικών, χρησιμοποιούμε ένα Inception Net [53], προεκπαιδευμένο σε βίντεο νοηματικής γλώσσας σε μια διάταξη CNN-LSTM-HMM [5]. Η πλήρης αρχιτεκτονική περιγράφεται παρακάτω.



Σχημα 1.10: Αρχιτεκτονική μοντέλου.

Στο πρόβλημα της αναγνώρισης δράσης, μια επιτυχημένη προσέγγιση είναι η εκπαίδευση ενός συνόλου GCNs με εισόδους τις θέσεις των αρθρώσεων και των διανυσμάτων οστών σε δύο ξεχωριστές ροές εισόδου. Αυτό υπογραμμίζει όχι μόνο τις δυνατότητες γενίκευσης του μοντέλου, αλλά και το πώς διαφορετικές αναπαραστάσεις των αρθρώσεων μπορούν να αλληλοσυμπληρώνονται και να αποδίδουν αυξημένη απόδοση. Έτσι, ακολουθούμε παρόμοια μέθοδο και εκπαιδεύουμε παράλληλα ένα μοντέλο με την ίδια αρχιτεκτονική αλλά με την παραμετροποίηση πόζας θ, έναν πίνακα 3D περιστροφής που εξάγεται από το ExPose ενσωματωμένο σε κάθε κορυφή / άρθρωση. Συνδυάζουμε τις πιθανότητες softmax που προβλέπονται από τα δύο μοντέλα και παρατηρούμε σημαντική μείωση του Word Error Rate (WER).

1.2.1.3 Προσαρμοσμένες συνελίξεις σε γραφήματα.

Η συνέλιξη σε γράφημα για τα δεδομένα σκελετού που περιγράφεται παραπάνω υπολογίζεται με βάση ένα προκαθορισμένο γράφημα, το οποίο ακολουθεί τη φυσική συνδεσιμότητα των αρθρώσεων στο ανθρώπινο σώμα και τα χέρια. Ωστόσο, εφαρμόζοντας ένα προσαρμοστικό γραφοσυνελικτικό επίπεδο μπορούμε να ξεπεράσουμε αυτόν τον περιορισμό και να βελτιστοποιήσουμε την τοπολογία του γραφήματος κατά τη διάρκεια της εκπαίδευσης [49]. Για να καταστήσουμε τη δομή του γραφήματος ευέλικτη, αλλάζουμε τη χωρική γραφική συνέλιξη σε:

$$f_k = \sum_{k=1}^{K_u} W_k (A_k + B_k + C_k) f_{in}$$

όπου,

A_k είναι ο κανονικοποιημένος πίνακας γειτνίασης.

B_k είναι ένας πίνακας γειτνίασης, που βελτιστοποιείται κατά τη διάρκεια της εκπαίδευσης. Αυτό επιτρέπει στο μοντέλο να διερευνήσει την ύπαρξη και τη δύναμη συνδέσεων μεταξύ απομακρυσμένων αρθρώσεων.

C_k είναι ένα γράφημα που εξαρτάται από τα δεδομένα και μαθαίνει μια μοναδική τοπολογία για κάθε δείγμα.

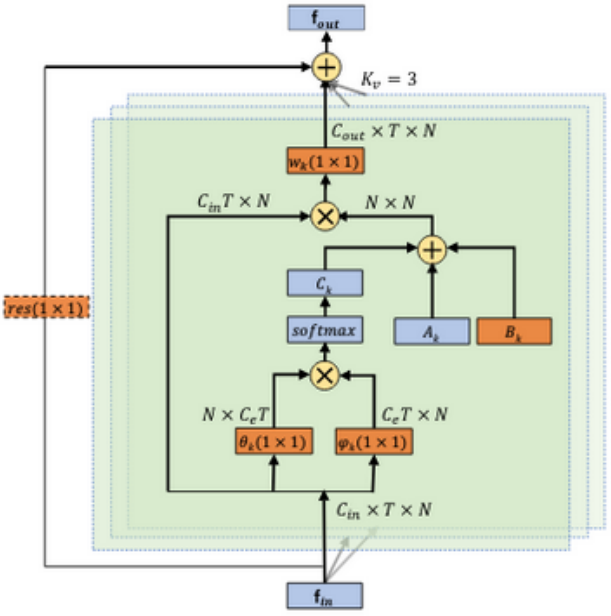
Πιο συγκεκριμένα, για να προσδιορίσουμε την ισχύ της σύνδεσης μεταξύ δύο κορυφών στο C_k, βασιζόμαστε στην ομοιότητά τους και εφαρμόζουμε την κανονικοποιημένη ενσωματωμένη συνάρτηση Gauss:

$$f(u_i, u_j) = \frac{e^{\theta(u_i)^T \theta(u_i)^T}}{\sum_{j=1}^{N} e^{\theta(u_i)^T \theta(u_i)^T}}$$

Αρχικά οι πίνακες χαρακτηριστικών μετατράπηκαν σε διάσταση Ce × T × N με δύο συναρτήσεις ενσωμάτωσης, δηλαδή θ και ϕ. Οι χάρτες χαρακτηριστικών αναδιαμορφώνονται σε έναν πίνακα N × CeT και έναν πίνακα CeT × N και πολλαπλασιάζονται για να αποκτήσουμε έναν πίνακα ομοιότητας C_k, του οποίου το στοιχείο C_{ijk} αντιπροσωπεύει την ομοιότητα της κορυφής i και της κορυφής j. Συγκεκριμένα, C_k υπολογίζεται ως εξής:

$$C_k = softmax({f_{in}}^T {W_{\theta k}}^T W_{\phi k} f_{in})$$

,όπου W_θ και W_ϕ είναι οι παράμετροι των συναρτήσεων ενσωμάτωσης.



Σχημα 1.11: Προσαρμοσμένη συνελίξη σε γραφήμα [49].

Επικεντρωνόμαστε στα πειράματά μας στο σύνολο δεδομένων Phoenix 2014T [4], το οποίο περιλαμβάνει συνεχή νοηματική γλώσσα από 9 διαφορετικούς νοηματιστές με λεξιλόγιο 1066 διαφορετικών λέξεων. Τα βίντεο είναι προσαρμοσμένα από καθημερινές ειδήσεις και προβολές καιρού από τον γερμανικό δημόσιο τηλεοπτικό σταθμό PHOENIX. Οι μεταφράσεις για αυτά τα βίντεο παρέχονται στη γερμανική ομιλούμενη γλώσσα. Όλα τα εγγεγραμμένα βίντεο έχουν 25 πλαίσια ανά δευτερόλεπτο και το μέγεθος των καρέ είναι 210 επί 260 pixels.

Η μέθοδος μας αποφέρει ελπιδοφόρα αποτελέσματα, ξεπερνώντας πολλές από τις προηγούμενες μεθόδους κατα σημαντικό ποσοστό και αποδίδει στο ίδιο επίπεδο με την τρέχουσα προσέγγιση του [60]. Παρατηρούμε ότι η εκμετάλλευση τόσο της θέσης άρθρωσεων

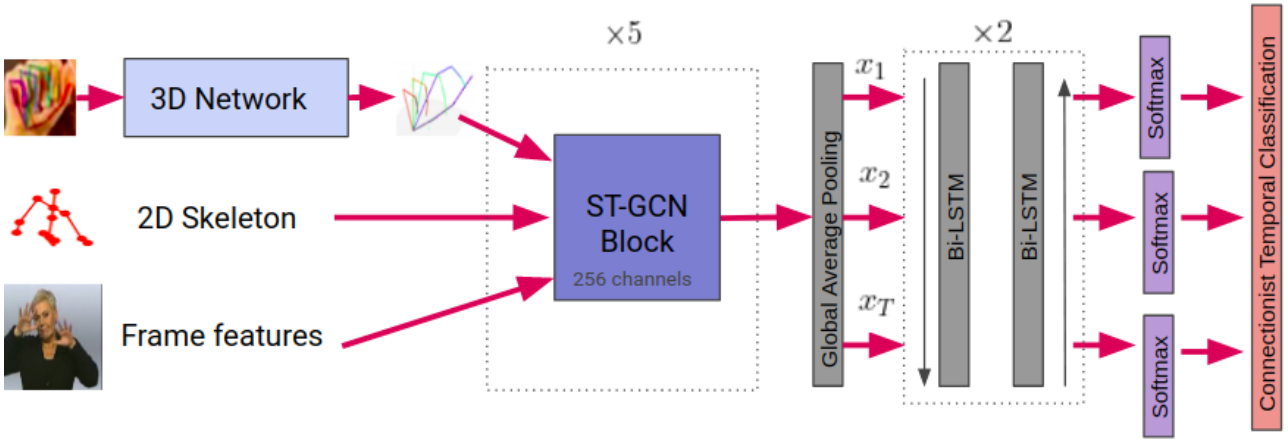
Πινακας 1.1: Σύγκριση απόδοσης του προσαρμοσμένου ST-GCN μας στο σύνολο δεδομένων Phoenix 2014T [4] σε dev και test set με διάφορες ροές χαρακτηριστικών.

Methods	WER(Dev)	WER(Test)
Our ST-GCN(2D-Skeleton + adaptive)	45.7 %	46.5 %
Our ST-GCN(3D-Skeleton + adaptive)	50.1 %	50.3 %
Our ST-GCN(ExPose + adaptive)	44.0 %	44.8 %
Our ST-GCN(appearance feats + 2D-3D skeleton)	23.5 %	23.8 %
Our ST-GCN(appearance feats + ExPose)	24.1 %	24.1 %
Our ST-GCN(appearance feats + ExPose+ 2D-3D skeleton)	22.6 %	22.8 %
Our ST-GCN(all modalities + GRU language model)	- %	22.4 %

(σκελετός) όσο και της 3D αναπαράστασης περιστροφής και παραμέτρων σχήματος (ExPose) μπορεί να βελτιώσει σημαντικά την ακρίβεια των προβλέψεων του μοντέλου. Εξετάζοντας προσεκτικά τα αποτελέσματά μας, καταλήγουμε στο συμπέρασμα ότι οι αποτυχημένες προβλέψεις των μοντέλων μας αφορούν λέξεις με παρόμοιο σημασιολογικό νόημα, όπως RISIKO-GEFAHR (ρίσκο-κίνδυνος), SUED-SUEDOST (νότια-νοτιοανατολικά), JUNI-JULI (Ιούνιος- Ιούλιος).

1.2.1.4 Μοντελοποίηση με LSTM

Τα χρονικά συνελικτικά στρώματα στα μπλοκ γραφικής συνέλιξης επιτυγχάνουν να εξάγουν μια αναπαράσταση χωροχρονικών γειτονιών και συλλαμβάνουν αποτελεσματικά βραχύχρονες κινήσεις. Ωστόσο, η μοντελοποίηση της μακροπρόθεσμης δυναμικής κίνησης είναι ζωτικής σημασίας στην αναγνώριση νοηματικής γλώσσας. Έτσι, επωφελούμαστε από τη δύναμη των LSTM, τα οποία μπορούν να επεξεργαστούν εξαρτήσεις μεγάλων αποστάσεων. Στην περίπτωσή μας εφαρμόζουμε έναν αμφίδρομο κωδικοποιητή LSTM με δύο επίπεδα και μια κρυφή κατάσταση διάστασης 256. Επιλέγουμε την αμφίδρομη ιδιότητα ώστε να εκμεταλλευόμαστε ταυτόχρονα το μελλοντικό περιβάλλον καθώς και το πρότερο πλαίσιο. Η είσοδος στον κωδικοποιητή είναι η χωροχρονική λανθάνουσα αναπαράσταση που εξάγεται από το τροποποιημένο δίκτυο ST-GCN και η έξοδος εισάγεται σε ένα γραμμικό στρώμα softmax για τις τελικές προβλέψεις. Οι τροποποιήσεις στην αρχιτεκτονική μας περιγράφονται παρακάτω:



Σχημα 1.12: Ολοκληρωμένη αρχιτεκτονική δικτύου με ST-GCN και Bi-LSTM επίπεδα.

Πινακας 1.2: Επίδοση ST-GCN-Bi-LSTM στο Phoenix 2014T dataset test set.

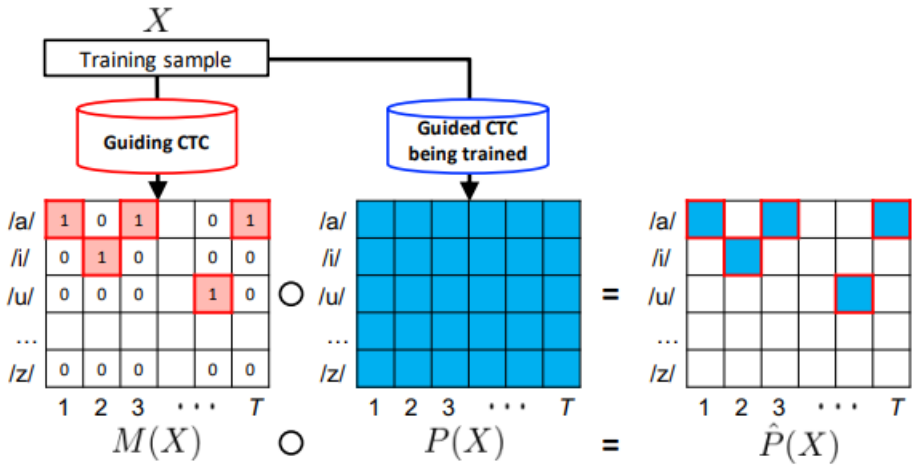
Methods	WER(Test)
ST-GCN-Bi-LSTM(appearance feats+2D-3D skeleton)	23.22 %
ST-GCN-Bi-LSTM(appearance feats+ExPose)	23.50 %

Όπως φαίνεται από τον παραπάνω πίνακα, το τροποποιημένο δίκτυο παρουσιάζει ανώτερη απόδοση τόσο στη διάταξη σκελετού + χαρακτηριστικών εμφάνισης όσο και στη διάταξη χαρακτηριστικών εμφάνισης + ExPose. Ωστόσο, η αξιοποίηση της posteriori σύντηξης πιθανοτήτων σε αυτήν την περίπτωση δεν είναι διαισθητική. Τα μοντέλα CTC εκπέμπουν πολύ αιχμηρές posteriori κατανομές όπου τα περισσότερα στιγμιότυπα εκπέμπουν το κενό σύμβολο και μερικά εκπέμπουν το σύμβολο στόχου. Ως αποτέλεσμα, τα μοντέλα CTC-LSTM παρουσιάζουν μη ευθυγραμμισμένους χρονισμούς ακίδων, γεγονός που καθιστά την σύντηξη αναποτελεσματική.

Προκειμένου να επιλυθεί αυτό το πρόβλημα, εφαρμόζουμε μια παρόμοια προσέγγιση με [32] όπου προτείνεται μια τεχνική που καθοδηγεί τους χρονισμούς ακίδων CTC με σκοπό την ευθυγράμμιση με εκείνους από ένα προ-εκπαιδευμένο μοντέλο CTC (το μοντέλο καθοδήγησης). Πιο συγκεκριμένα, όταν εκπαιδεύουμε ένα μοντέλο CTC, εκτός από την ομαλοποιημένη συνάρτηση απώλειας CTC, προσθέτουμε έναν όρο απώλειας που καθοδηγεί τις αιχμές από το μοντέλο που εκπαιδεύεται για να συμπίπτουν με εκείνες του μοντέλου καθοδήγησης. Με αυτόν τον τρόπο τα μοντέλα που καθοδηγούνται από το ίδιο μοντέλο καθοδήγησης παρουσιάζουν ευθυγραμμισμένους χρονισμούς ακίδων.

Κατά την εκπαίδευση του καθοδηγούμενου μοντέλου CTC, οι posteriori πιθανότητες για κάθε χρονική στιγμή που προβλέπονται από το μοντέλο καθοδήγησης μετατρέπονται σε μάσκα $M(X)$ ρυθμίζοντας στο 1 το σύμβολο εξόδου με την υψηλότερη πιθανότητα και στο 0 τα άλλα σύμβολα σε κάθε χρονική στιγμή. Όταν το κενό σύμβολο παρουσιάζει την υψηλότερη πιθανότητα, θέτουμε 0 σε αυτό το χρονικό σημείο. Αυτή η μάσκα εφαρμόζεται στα σύμβολα εξόδου που εκπέμπει το μοντέλο καθοδήγησης CTC.

Πιο συγκεκριμένα, κατά τη διάρκεια της εκπαίδευσης οι posteriori πιθανότητες P(X) προβλέπονται από το καθοδηγουμενο μοντέλο. Μέσω ενός πολλαπλασιασμού στοιχείο προς στοιχείο της μάσκας και των πιθανοτήτων, λαμβάνουμε $\hat{P}(X) = M(x) \odot P(X)$. Με τη μεγιστοποίηση του αθροίσματος αυτού, μπορούμε να συγχρονίσουμε τους χρονισμούς ακίδων CTC του καθοδηγούμενου μοντέλου με εκείνους του μοντέλου καθοδήγησης CTC.



Σχημα 1.13: Διάγραμμα της προτεινούμενης καθοδηγούμενης CTC εκπαίδευσης [32].

Στα πειράματα νοηματικής γλώσσας χρησιμοποιούμε το μοντέλο που έχει εκπαιδευτεί σε τρισδιάστατους σκελετούς και χαρακτηριστικά εμφάνισης ως μοντέλο καθοδήγησης και εκπαιδεύουμε ένα δεύτερο μοντέλο καθοδήγησης που χρησιμοποιεί τα δεδομένα ExPose και χαρακτηριστικά εμφάνισης ως είσοδο με τη μέθοδο που περιγράφεται παραπάνω. Τα αποτελέσματα της posteriori σύντηξης παρουσιάζονται παρακάτω:

Πινακας 1.3: Απόδοση του posteriori συνδυασμού πιθανοτήτων στο Phoenix2014T test set.

Methods	WER(Test)
2x ST-GCN-BLSTM	21.9 %

Παρατηρούμε μια σημαντική βελτίωση σε σχέση με τα μεμονωμένα μοντέλα, η οποία ήταν αναμενόμενη.

Chapter 2

Introduction

Humans have strived for decades to provide machines with the ability to "think" and formulate an understanding of their 3D environments and peoples' actions and expressions. Computer-vision based gesture recognition and pose estimation methods aim to facilitate effective and natural human-computer communication and interactions. Humans tend to naturally use hand gestures in their communication processes in order to convey feelings and information. Gesture recognition allows computers to capture and interpret human gestures and execute commands.

Research in gesture recognition has been developing for decades. In the previous years wearable hand gesture devices and sensors and later traditional computer vision methodologies, such as feature descriptors and support vector machines dominated the field. The advent of deep learning revolutionized the landscape and allowed for new powerful approaches in this domain that draw on the availability of large-scale datasets. In this thesis we tackle the problem of sign language recognition, which lies at the intersection of gesture recognition and hand pose estimation.

2.1 Problem Definition

Sign language is the language of the Deaf and hard of hearing and constitutes their main medium of communication. Sign language recognition is a challenging and complex computer vision task of multi-articulatory nature, which fuses visual and language modeling. Multiple cues, such as facial expression, hand shape and pose, and body movements are combined to effectively convey information. Sign language models have to extract a rich spatio-temporal representation from an input video and understand how a signer moves within a 3D signing space. In this aspect the task of SLR can be considered a sub-problem

of action and gesture recognition. However, the goal of an SLR system is to predict the sequence of glosses, performed by the signer. SL is a complete language, with all the nuances and subtleties of a spoken language, a fact which renders SLR a complex linguistic modeling task. Computer vision research has focused extensively on isolated sign language recognition. With the emergence of large scale recognition corpora, research has shifted primarily to continuous sign language recognition.

FIGURE 2.1: A signer's facial expression, handshape and upper body movements play an important role in the articulation of the sign.

3D Hand Pose Estimation is a sub-discipline of Computer Vision which aims to extract a 3D representation of the joints of the human hand. The location, orientation and articulation of the hand in the three dimensional space is useful for a plethora of applications, including sign language recognition in which subtle finger movements play a major role. Many of the previous works employed multiple cameras and depth information to infer 3D Pose but in this thesis we focus on novel methods that utilize monocular RGB images and can generalize effectively to unseen domains.

2.2 Contribution

Our contribution in this thesis is twofold. Firstly, recognising the importance of the knowledge of the detailed trajectory of arms and fingers we propose methods that are successful in extracting the 3D body and hand skeleton joints of the signers and can generalize to videos in the wild.

Secondly, after obtaining a richer representation of the human skeleton we investigate how full body shape and pose can be exploited and effectively fused with visual features in order to perform continuous sign language recognition. We conduct our experiments

and report competitive performance on par with the current state-of-the-art in the widely researched RWTH Phoenix 2014T dataset.

- Inspired by successful approaches on 3D body and hand pose estimation we implement a deep linear architecture with geometric constraints for 3D hand pose "lifting" and thoroughly test its performance on two 3D hand pose datasets. We also measure the generalization power of our method and provide qualitative examples of its predictions when in-the-wild sign language frames are used as an input.
- To the best of our knowledge this thesis is one of the first works to fuse 3D full body shape and pose parameters and exploit a rich 3D representation of the human body in the task of sign language recognition.
- We leverage the power of graph convolutional networks, which have not been explored thoroughly in continuous sign language recognition. 3D joint position and axis-angle rotation representation are embedded in the graph nodes and fused with frame appearance features. An LSTM encoder is incorporated after the graph convolutional layers for temporal modeling, which yields increased performance.
- The contribution of a late fusion weighted technique is examined, where the posteriors of the different cues models are synchronized and fused for gloss prediction.
- We report competitive results, surpassing many state-of-the-art approaches on the challenging RWTH Phoenix 2014T dataset and compare our method with strong baseline sign language transformer architectures and decoders with different attention mechanisms.

2.3 Thesis Outline

In Chapter 3 we present two frameworks: OpenPose and ExPose, which we used extensively for feature extraction and provide theoretical background on basic deep learning architectures.

In Chapter 4 we present representative, previously published works in 3D Hand and Body Pose Prediction and Sign Language Recognition.

In Chapter 5 we present our experiments on Isolated Sign Language Recognition, which are motivated by successful skeleton-based approaches in action and gesture recognition.

In Chapter 6 we analyze our approach on 3D Hand and Body Pose estimation and study the qualitative and quantitative results we obtain.

In Chapter 7 we present our method on Continuous Sign Language Recognition and compare it with strong baseline approaches. We further analyze our results.

In Chapter 8 we summarize our efforts and propose future directions.

Chapter 3

Background on Deep Learning Architectures and Pose Estimation

In this chapter we briefly analyze common architectural blocks in state-of-the-art models for Computer Vision and Natural Language Processing. Additionally, we provide background information on pose estimation and discuss both modern and traditional computer vision approaches on this task.

3.1 Deep Learning Architectures

3.1.1 Recurrent Neural Network Architectures

In this section we examine the use of recurrent neural networks [47] and LSTM neurons (Long Short Term Memory Neurons) [18], which are effective in modeling long-term contextual information of temporal sequences. In traditional fully connected neural networks, all the inputs and outputs are independent of each other. However, when dealing with sequential data context is vital, namely the network ought to "remember" information learnt from prior input(s) while generating output(s).

The primary difference between RNNs and feedforward networks lies in the existence of feedback loops, that produce the recurrent connection in the unfolded network.

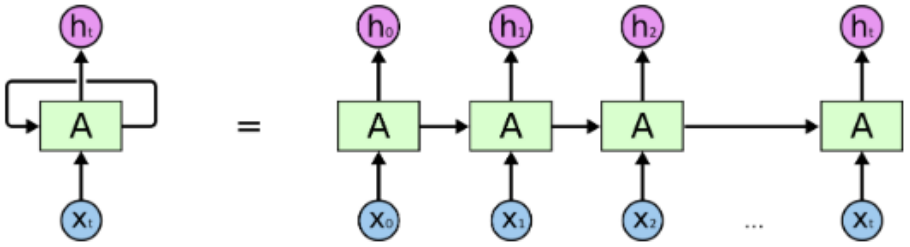


FIGURE 3.1: Unrolled RNN

Given an input sequence $x = (x^0, ..., x^{T-1})$, the hidden states of a recurrent layer $h = (h^0, ..., h^{T-1})$ and the output of a single hidden layer RNN $y = (y^0, ..., y^{T-1})$ can be computed as follows:

$$h^t = H(W_{xh}x^t + W_{hh}h^{t-1} + b_h)$$

$$y^t = O(W_{ho}h^t + b_o)$$

, where W_{xh}, W_{hh}, W_{ho} denote the connection weights from the input layer x to the hidden layer h , the hidden layer h to itself and the hidden layer to the output layer y respectively, b_h and b_o are two bias vectors, $H()$ and $O()$ are the activation functions in the hidden layer and the output layer.

The hidden state vector captures the relationship that neighbors might have with each other in a serial input and it is updated at every step to reflect the latest part of the input sequence and the context until this time step.

3.1.1.1 Bidirectional Neural Network (Bi-RNN)

In many cases it is desirable to overcome the limitations of RNNs and utilize information from both past (backwards) and future (forward) states simultaneously. The premise of Bidirectional Recurrent Neural Networks (BiRNNs) is to separate the state neurons of a regular RNN in a part that that encodes input flow in the positive time direction and a part for the negative time direction. These two recurrent hidden layers share the same output layer.

3.1.1.2 Long Short Term Memory Network (LSTM)

LSTM is an advanced RNN architecture, which overcomes the vanishing gradient problem and effectively captures long term dependencies. Vanishing gradients is when the gradients of loss functions become too small (approach zero) and the network becomes

increasingly hard to train, as the weights and biases of the initial layers are not updated effectively with the training sessions. In LSTM networks non linear units in RNN are replaced by LSTM memory blocks.

The LSTM memory block contains one self connected memory cell c_t and three multiplicative gates, e.g the input gate i_t, the forget gate f_t and the output gate o_t. It has the ability to remove or add information to the cell state, carefully regulated by the gates. The input gate layer determines which values we'll update and the forget gate governs the flow of information, by deciding which information will be excluded. The output gate controls the amount of information passed from the cell to the output. The memory cell has a self connected edge of weight 1, so that the gradient does not vanish or explode. The activation of the memory cell and the three gates are as follows:

$$i^t = \sigma(W_{xi}x^t + W_{hi}h^{t-1} + W_{ci}c^{t-1} + b_i)$$

$$f^t = \sigma(W_{xf}x^t = W_{hf}h^{t-1} + W_{cf}c^{t-1} + b_f)$$

$$c^t = f_t c_{t-1} + i_t tanh(W_{xc}x^t + W_{hc}h^{t-1} + b_c)$$

$$o^t = \sigma(W_{xo}x^t + W_{ho}h^{t-1} + W_{co}c^t + b_o)$$

$$h^t = o_t tanh(c^t)$$

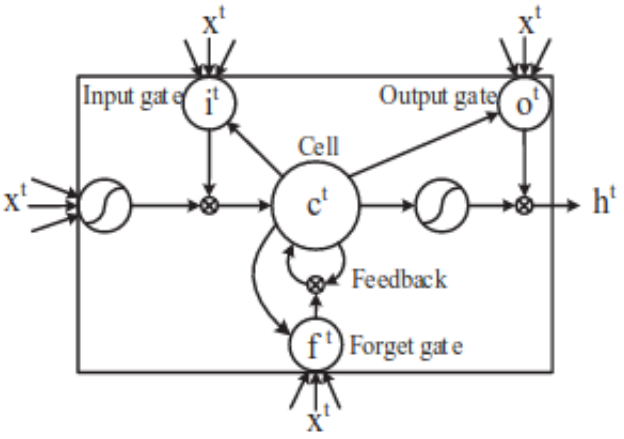


FIGURE 3.2: LSTM block with one cell

3.1.1.3 Gated Recurrent Units

Gated Recurrent Units [10] limit the parameter space of the LSTMs through the introduction of a separate context vector, and by reducing the number of gates to 2, a reset gate, r and an update gate, z. The objective of the reset gate is to filter aspects of the previous hidden state according to their relevance to their current concept. This is achieved by an element wise multiplication of r_t with the previous hidden state h_{t-1} and the result is used to produce an intermediate state representation as follows:

$$r_t = \sigma(U_r h_{t-1} + W_r x_t)$$

$$h_{\tilde{t}} = \tanh(U(r_t \odot h_{t-1}) + W_{xt})$$

The objective of the update gate, z, is to control how much information from the current memory content $h_{\tilde{t}}$ and the previous steps h_t will be retained. Specifically, the values of z are used to interpolate between the new and the old hidden state.

$$h_t = (1 - z_t)h_{t-1} + z_t h_{\tilde{t}}$$

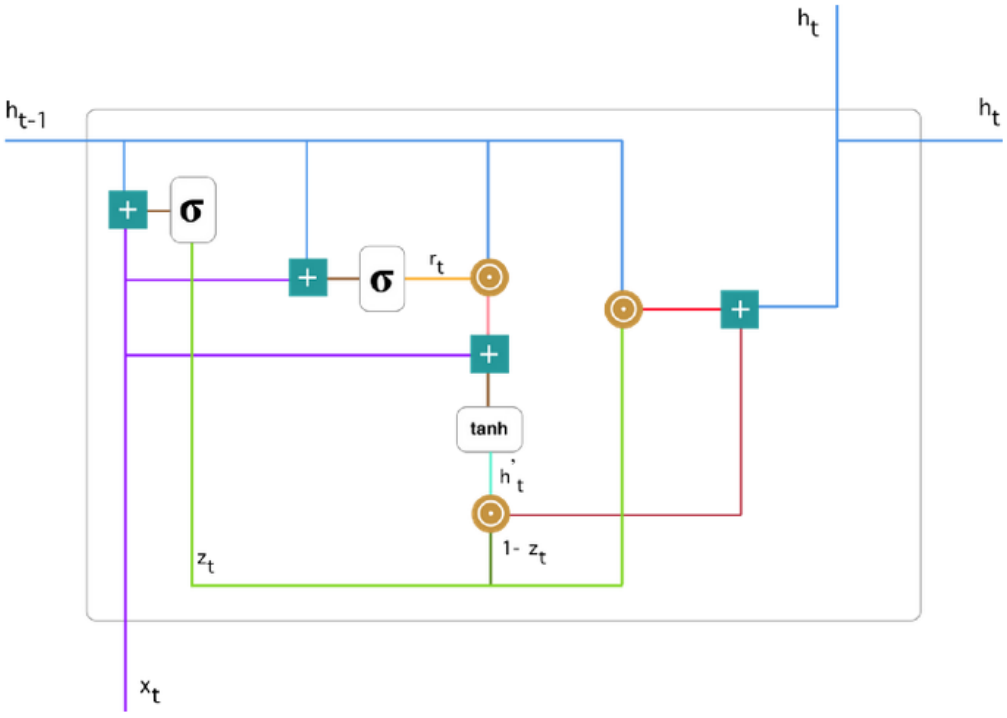


FIGURE 3.3: Architecture of GRU block.

3.1.2 Convolutional Neural Networks

Traditional computer vision methods were dominated by hand-engineered filters. Morphological operators, such as erosion, dilation, opening and closing as well as laplacian kernels were extensively used for detecting sub-patterns and basic image features, including edges and boundaries. With the advent of deep learning, Convolutional Neural Networks [33] emerged, which have the ability to learn data-dependent filter weights.

In fully connected networks each neurons' activation can be expressed as a weighted function of the activations of all neurons in the previous layer. Thus, transforming an input image to a vector of pixels and utilizing a fully connected architecture would be computationally prohibitive. Convolutional networks introduce sparse connections, by allowing each neuron to attend to a small neighborhood of the input. In this sense CNNs possess the property of translation invariance, meaning they produce the same response, regardless of shifts and variations in viewpoint. The underlying inspiration behind the convolution operator is the biological mechanism of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field, namely the receptive field

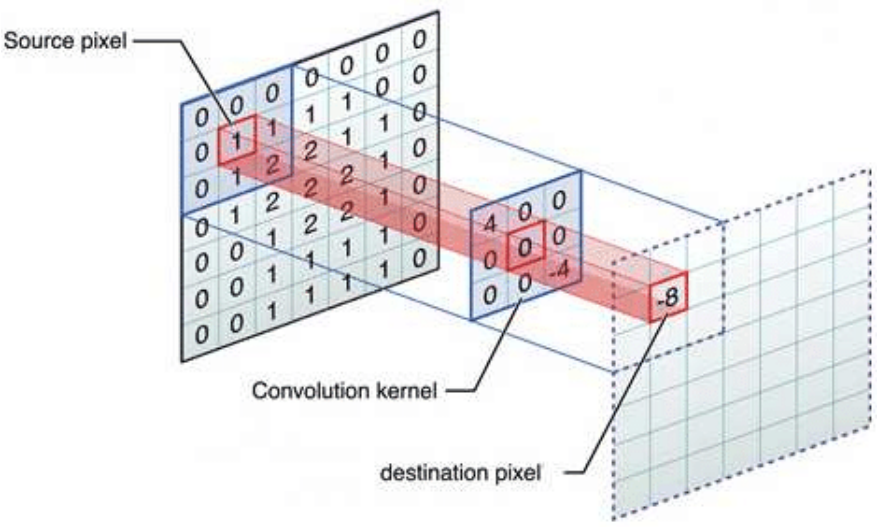


FIGURE 3.4: Convolution operation.

In each layer of a convolutional network filters/kernels traverse the input image by performing a matrix multiplication operation between the kernel parameters and the image segment over which the kernel is hovering. The objective is to extract meaningful high-level features and capture the spatial dependencies of the input. The first layers of a convolutional network extract low-level features, such as edges, colour and lines while the top layers produce the richest possible category specific features representation. Pooling layers are usually incorporated in deep CNN architectures. Pooling layers gradually decrease the spatial extent of the network. Max Pooling and Average Pooling return the

maximum and the average value respectively from the section of the image covered by the kernel. A simple CNN architecture is outlined below.

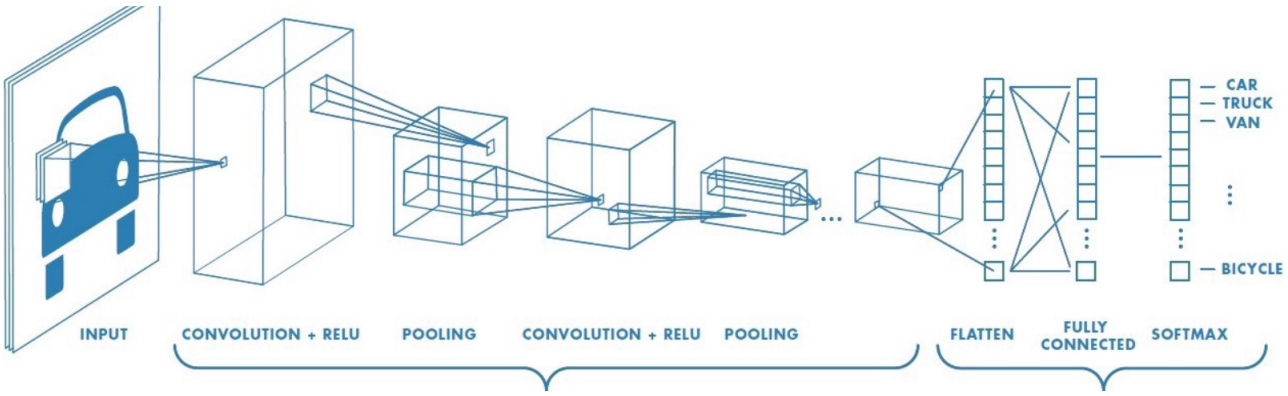


FIGURE 3.5: A simple CNN architecture.

3.1.3 Transformer Networks

Transformers [55] utilize attention mechanisms to draw global dependencies between inputs and outputs. They have achieved state-of-the art results in many language modeling and understanding tasks, without employing recurrence. The main premise behind the success of transformers is self-attention. Self-attention is an attention mechanism that relates different positions of a single sequence in order to compute a representation of the sequence. In the calculation of self-attention the embedding of each input is multiplied by three learned matrices Q, K and V and a Query vector, a Key vector, and a Value vector are created respectively. The score is calculated by taking the dot product of the query vector with the key vector of the respective position in the input. The matrix of outputs is computed as:

$$Attention(Q, K, V) = softmax(\frac{Q \cdot K^T}{\sqrt{d_k}}) \cdot V$$

,where d_k is the dimension of the queries and keys

The Transformer refines self-attention by introducing multi-head self attention. The queries, keys and values are linearly projected h times with different, learned linear projections to d_k, d_k and d_v dimensions, respectively. This allows the model to attend to different representation sub-spaces at different positions.

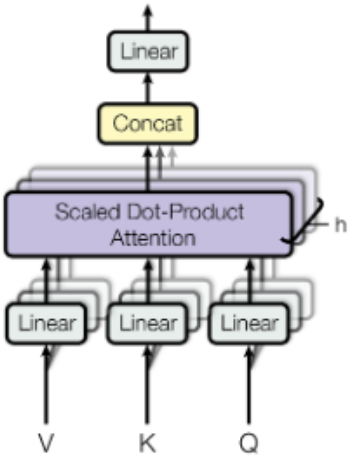


FIGURE 3.6: Multi-head Self-Attention in Transformer [55]

The transformer follows an encoder-decoder structure. The encoder maps an input sequence to a sequence of representations in a latent space. Consequently, the decoder generates an output sequence one element at a time by relating the encoded representation with the output at each time step. The encoder in the original transformer architecture consists of 6 layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a position-wise fully connected feed-forward network.

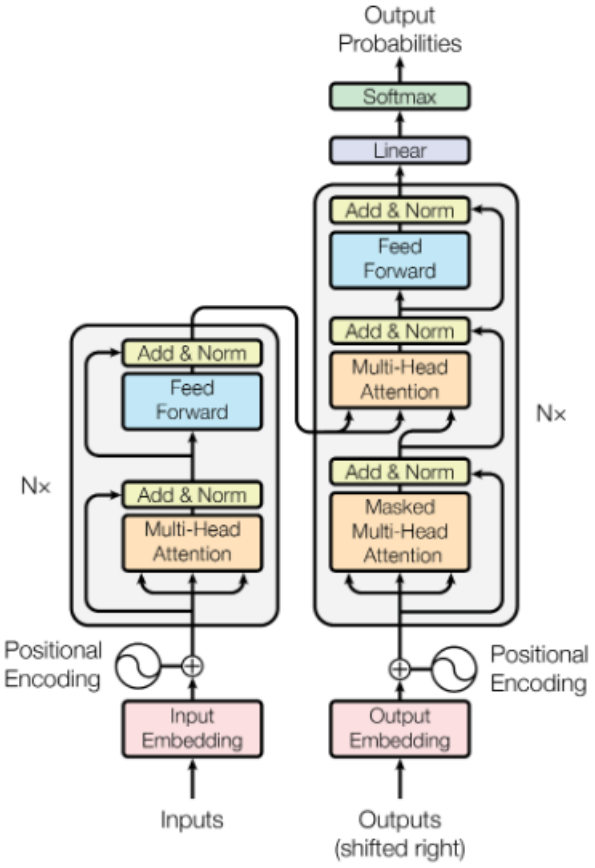


FIGURE 3.7: Transformer Encoder-Decoder Architecture [55].

3.1.4 Graph Convolutional Networks

Graph Convolutional Networks extend the notion of convolution to signals that reside in the graph domain. They are used to solve problems in which input data can be expressed in the form of a graph, such as social networks and protein structures. In our case the 3D human skeleton can be modeled by a graph G(V,E) where the nodes V correspond the body and hand joints and the edges represent the bones.

In order to familiarize the reader with the concept of GCNs we provide some introductory information. An undirected unweighted graph G is represented by a pair (V,E) where V is the set of vertices $v_1, v_2, ..., v_n$ and E is the set of edges of the graph. A pair $i, j \epsilon\ E$ if vertices v_i and v_j are adjacent. The most commonly known operator of a graph G is the adjacency matrix. It is a symmetric matrix defined as:

$$A(i,j) = \begin{cases} 1 & if(i,j)\ \epsilon\ E \\ 0 & otherwise \end{cases}$$

The degree matrix is a diagonal matrix with entries the degree of the vertex defined as $di = \sum_j 1 : (i, j)\ \epsilon\ E$. Finally, the Laplacian operator of an unweighted graph can defined as $L = D - I$. The eigendecomposition of the Laplacian matrix provides meaningful information about its structure, such as the volume, the diameter and the vertex degrees of the graph.

Spectral convolutions on graphs are defined as the multiplication of a signal with a filter $W_\theta = diag(\theta)$ in the Fourier domain, i.e.

$$W_{theta} * x = UW_\theta U^T x$$

,where V is the matrix of eigenvectors of the normalized graph Laplacian $L = D^{1/2}AD^{-1/2}$ and $V^T x$ is the graph Fourier transform of x. W_θ can be perceived as a function of the eigenvalues of L, i.e. $W(\Lambda)$. It was suggested in [15] that $W_\theta(L)$ can be approximated by an expansion in terms of Chebyshev polynomials $T_k(x)$ up to Kth order: $W_\theta(\Lambda) = \sum_{k=0}^{K} = \theta_k T_k(\Lambda)$ with a rescaled $\Lambda = \frac{2}{\lambda_{max}}\Lambda - I_N$ where λ_{max} denotes the largest eigenvalue of L. Thus, the convolution of a signal x with a filter W_θ becomes: $W_\theta * x = \sum_{k=0}^{K} = \theta_k T_k(\Lambda)x$ If we limit the convolutional operator to K=1 and approximate $\lambda_{max} = 2$ the above equation can be simplified to :

$$W_\theta * x = \theta(I_N + D^{1/2}AD^{-1/2})x$$

This defines the propagation rule in a graph convolutional network [26].

Given a graph $G(V, E, A)$ where V is the set of nodes, E is the set of edges and A is the adjacency matrix a GCN takes as an input a feature matrix $N \times F^0$ X where N denotes the number of nodes and F^0 the input features for each node.

A hidden layer in the GCN can be written as follows:

$$H^i = F(H^{i-1}, A)$$

where $H^0 = X$ and H^i corresponds to a $N \times F^i$ feature matrix and f symbolizes the propagation rule. At each layer features are aggregated to form the next layers' features. The propagation rule is

$$f(H^l, A) = \sigma(AH^{(l)}W^{(lj)})$$

where σ is a non-linear activation function such as the ReLU function and W^i is the weight matrix for layer i. The weight matrix has dimension $F^i \times F^{i+1}$.

Graph Convolutional Networks have demonstrated increased performance in a wide range of tasks. However, for the purpose of this thesis we focus on their uses in action and gesture recognition tasks.

3.2 Pose and Shape Estimation

The extraction of 2D and 3D visual pose and shape features has been a long standing problem in computer vision. Traditional approaches rely on handcrafted features and leverage the power of feature descriptors to create a representation of a gesture/sign. We briefly discuss some widely employed feature descriptors that operate on RGB images.

- Histogram of Oriented Gradients (HOG)/ Histogram of oriented Optical Flow (HOF) : Was introduced by [13]. It is based on the calculation of the histograms of spatial gradients and optical flow in the spatio-temporal neighborhood of the selected points of interest.
- HOG3D: It was proposed by [27] and is based on 3D oriented gradient histograms. The 3D HOG feature descriptor can describe both the shape and motion feature with a spatio vector. In accordance with the HOG/HOF features, each spatiotemporal block is divided into nx x ny × nt cells. The individual histograms of the gradients for each cell are pooled in the same vector, which after normalization leads to the final descriptor.

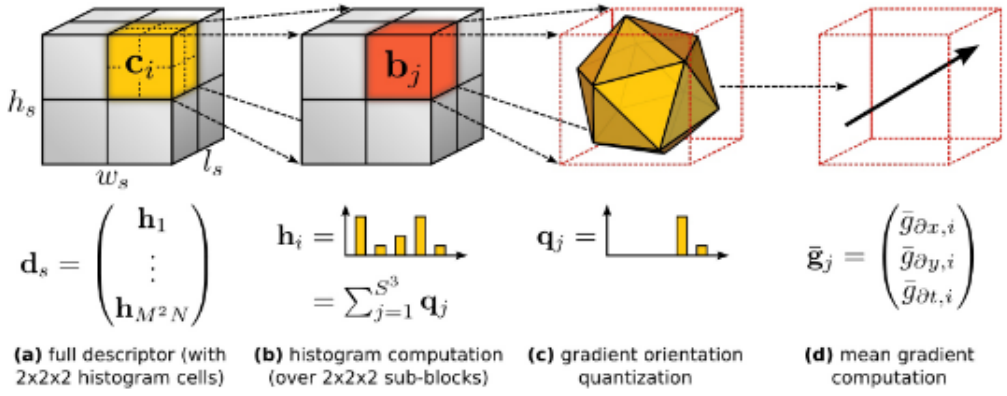


FIGURE 3.8: Illustration of the 3D HOG descriptor [28].

[42] extracts HOG features from segmented images for both RGB and depth modality in order to train unimodal gesture-word models. The hands are segmented via hand tracking and threshold depth segmentation.

With the advent of deep learning many novel methods for pose and shape estimation have emerged that can generalize to unseen frames and are robust to noise. We focus on the following two.

3.3 OpenPose Framework

OpenPose [6, 7, 51] is one the most widely used frameworks for full body 2D skeleton joint prediction. An RGB monocular image is used as an input and the model jointly predicts hand, body and face 2D anatomical keypoints. More specifically, the output is 21 keypoint locations, corresponding to each hand, 25 keypoints for the body and 70 keypoints for the face. Each keypoint is represented by a three dimensional vector which consists of the x and y pixel coordinates of the keypoint location in the frame and the confidence c of the prediction.

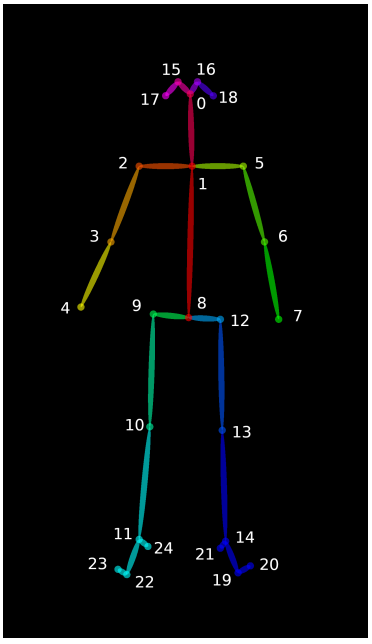


FIGURE 3.9: 25 body keypoints

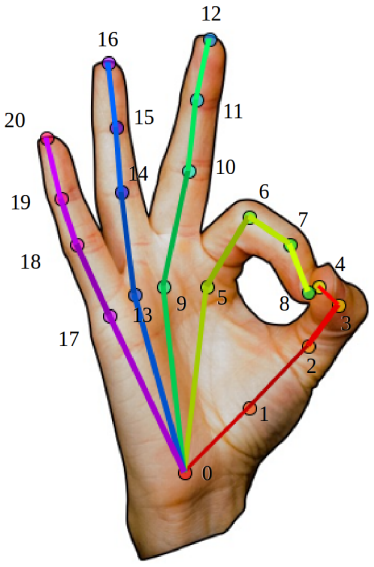


FIGURE 3.10: 21 hand keypoints

3.3.1 Model Architecture

The model is a two-branch multi-stage Convolutional Neural Network. The input image is fed to a CNN, that extracts image features, namely the first 10 layers of VGG-19. At each stage the top branch passes the image features through a series of convolutional layers in order to obtain a set of confidence map S of body part locations. The bottom branch, also consisting of a stack of convolutions predicts the part affinity fields L, which are a set

of flow fields that encode pairwise relationships between body parts. In each subsequent stage the predictions from both branches along with the original image features, are concatenated and used to produce more refined predictions. Formally,

$$S = (S_1, S_2, ..S_j)$$

$$S_j \epsilon R^{w \times h}$$

$$L = (L_1, L_2, ..L_j)$$

$$L_j \epsilon R^{w \times h \times 2}$$

$$S^t = p^t(F, S^{t-1}, L^{t-1})$$

$$L^t = \phi^t(F, S^{t-1}, L^{t-1})$$

where:

p is a function that represents the CNN with input F

ϕ is a function that represents the CNN with input F

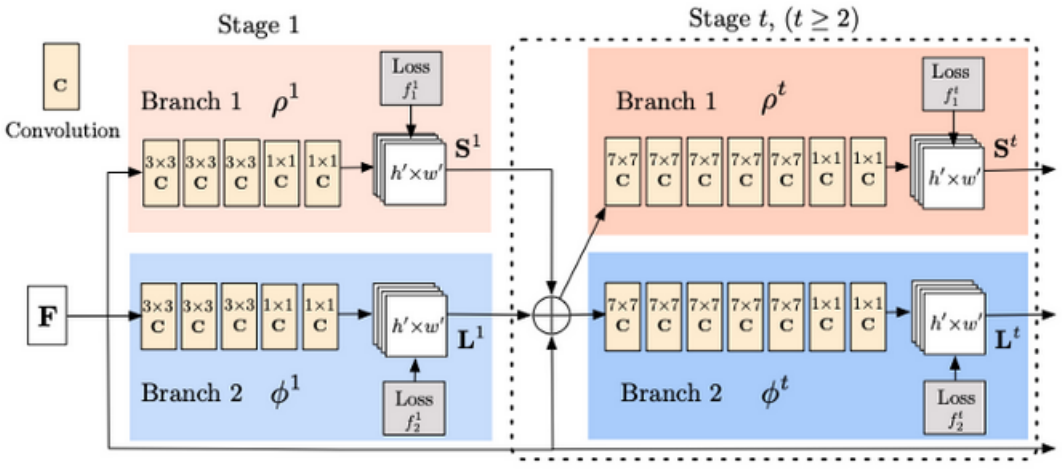


FIGURE 3.11: Architecture of the OpenPose network [6].

Two loss functions are applied at the end of each stage, one at each branch respectively. A standard L2 loss between the estimated predictions and ground truth maps and fields is used. Formally,

$$f_s^t = \sum_{j=1}^{J} \sum_{p} W_p \| S_j^t(p) - S_j^* \|_2^2$$

$$f_l^t = \sum_{c=1}^{C} \sum_{p} W_p \| L_c^t(p) - L_c^* \|_2^2$$

where:

p is the pixel location

S(p) is the confidence score at image location p

L(p) is directional vector for limb c at image location p

W(p) is a binary mask when the annotation is missing

3.4 ExPose framework

ExPose [12] is a framework, which directly regresses the 3D body, face, and hands from an RGB image in a SMPL-X format. SMPL-X [41] is a unified, body generative model that captures the overall shape and pose of the body. Formally, SMPL-X is defined by a differentiable function $M(\theta, \beta, \psi) : R^{|\theta| \times |\beta| \times |\psi|}$ that outputs a triangulated mesh with N = 10,479 vertices. This function is parameterized by the pose vector θ, the joint body shape parameters β and the facial expression parameters ψ.

ExPose parameterizes both shape β ϵR^{10} and expression ϕ ϵR^{10} using 10 coefficients derived from the PCA space. The articulation of the body, hands and face is modeled by a pose vector θ $\epsilon R^{J \times D}$ where D is the rotation representation dimension. In our case J = 53 and D = 6. More specifically, 53 joints are extracted, namely 22 body-pose joints, 15 joints per hand, and 1 for jaw, with a rotation representation dimension equal to 6.

3.4.1 ExPose Architecture

ExPose takes advantage of an attentive architecture to regress full body parameters. Bounding boxes of the body, face and hands are extracted and fed to separate networks so that the predictions can be refined. A body crop, derived from an affine transformation from the high resolution image is used as input to a network g in order to derive a first set of SMPL-X parameters. Initially, convolutional image features are encoded by a Resnet-50 network, pretrained on ImageNet. Consequently, the feature maps are passed

to an iterative 3D regression module, which consists of three fully connected layers. The objective of this module is to regress parameters that represent the 3D reconstruction of a human body that minimizes the joint reprojection error. This is achieved via an iterative error feedback loop. Additionally, an adversarial discriminator is employed, which acts as a data-driven prior that guides the 3D inference and penalizes unnatural poses. In the second stage the posed joints are projected on the image and used for the extraction of a bounding box for each hand and the face. Using these boxes, affine transformations are computed to extract higher resolution hand and faces images via spatial transformers. The hand and face images are fed to a part-specific hand and face network respectively, to refine the corresponding parameter predictions. For the face and hand sub-networks a ResNet18 is utilized.

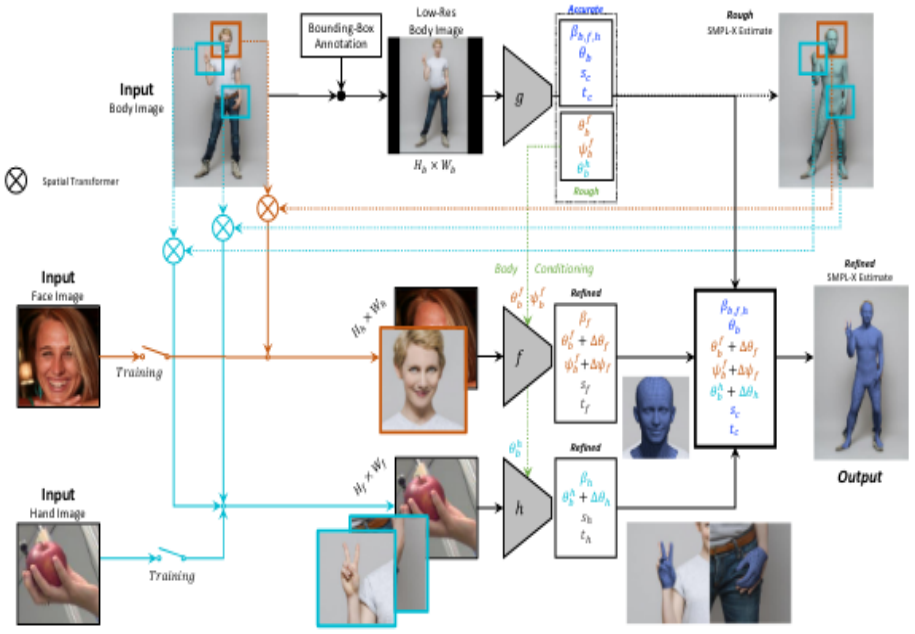


FIGURE 3.12: Architecture of the ExPose network [12]

We present examples of the 3D mesh produced by ExPose when frames from the Phoenix2014T dataset [4] are used as input.

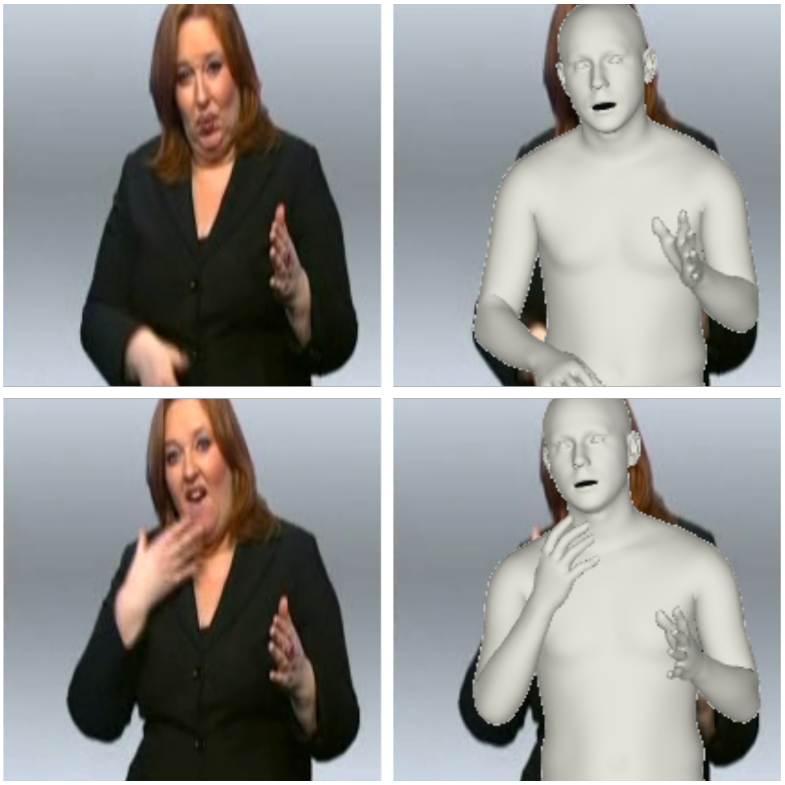


FIGURE 3.13: Examples of 3D body renderings, produced by ExPose.

Chapter 4

Previous Works

In this chapter we provide a brief overview of current state-of-the-art methods on 3D hand pose estimation and sign language recognition.

4.1 Previous works on 3D Hand Pose estimation

Hand-based articulation plays a primary role in Sign Language and mutiple signs exhibit similar skeletal motion patterns. Thus, it is vital to seek methods that enrich hand motion and structure information and provide models with meaningful knowledge about the trajectory of hands and fingers in the three dimensional plane. In the first part of our thesis we shifted our focus towards 3D pose estimation methods, that can effectively lift 2D coordinates to the 3D space.

3D Pose estimation is a long standing problem, which has given birth to a plethora of important applications. 3D hand pose estimation is particularly demanding because of many ambiguities, strong articulation, and heavy self-occlusion. Following the emergence of RGB-D sensors, many efforts have been devoted to 3D hand pose estimation through depth sensor and RGB input. However, in the majority of SL corpora and in real-world settings, signs are not recorded with depth sensors and depth information is unavailable. Thus, recent works focus mostly on 3D hand pose estimation from monocular RGB images. One of the first works that tackled this problem was [61], who adopted a three stage pipeline that performs hand segmentation, 2D joint generation, and finally 3D joint prediction. More specifically, the hand is localized within the image by a segmentation network (HandSegNet). Accordingly to the hand mask, the input image is fed to an encoder-decoder pose network, which predicts score maps containing information about the likelihood that a certain keypoint is present at a spatial location. Subsequently,

a PosePrior network estimates the the 3D structure conditioned on the score map, by predicting coordinates within a canonical frame and additionally estimating the transformation into the canonical frame. The PosePrior networks' architecture consists of two parallel processing streams. They first process the stack of score maps with a series of convolutions and consequently the feature representation are fed to two fully-connected layers. Finally, a fully-connected layer with linear activation yields the estimations for viewpoint and canonical coordinates. The complete network architecture is outlined below.

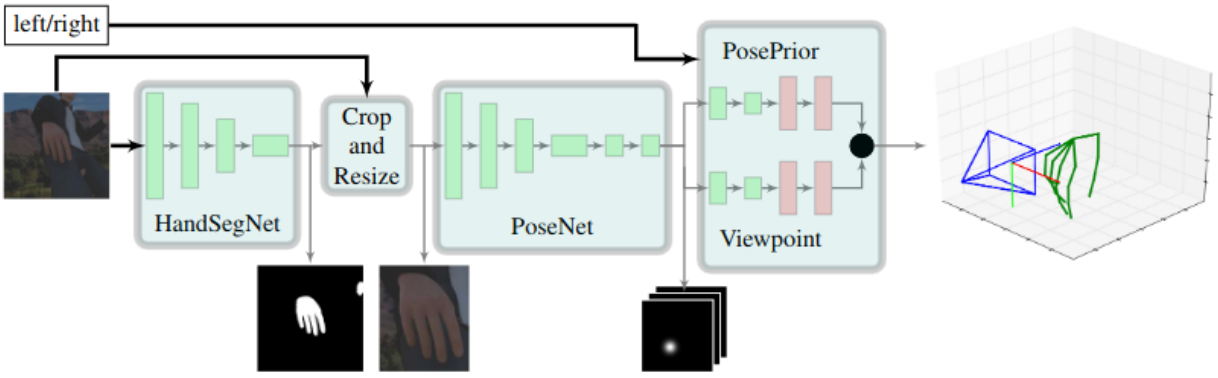


FIGURE 4.1: The main building blocks of the 3D Pose estimation network [61].

More recent works have emerged that take advantage of adversarial models. The work in [37] proposes a cycle-consistent generative adversarial network (CycleGAN) which transforms synthetic 3D annotated hand images into real looking ones, whose statistical distribution matches real-world hand images. Cycle GAN exploits the property that translation should be "cycle consistent", to learn various image-to-image translation tasks with unpaired examples. Specifically, if we translate from one domain to the other and back we should arrive at our starting point. Forward cycle-consistency loss can be defined as $x \rightarrow G(x) \rightarrow F(G(x)) \approx y$, and backward cycle-consistency loss as $y \rightarrow F(y) \rightarrow G(F(y)) \approx x$ where G and F are mapping functions between the two domains X and Y. In this work mappings from synthetic to real images and from real to synthetic images are learned. An additional geometric consistency loss is incorporated that ensures that the images maintain the hand pose during image translation.

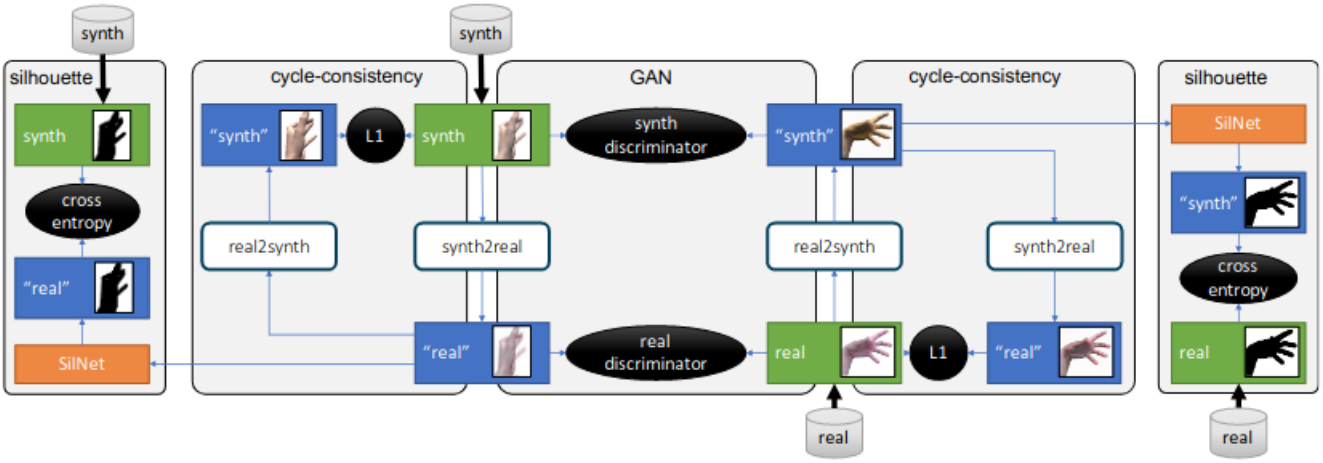


FIGURE 4.2: Network architecture of GeoConGAN [37].

The resulting annotated images produced by the proposed GAN are trained via a convolutional neural network (CNN) regressor for 2D and 3D hand joint predictions. More specifically a residual network consisting of 10 residual blocks that is derived from the ResNet-50 [16] architecture is utilized and an additional projection layer is incorporated, which performs orthographic projection of intermediate 3D predictions, from which 2D Gaussian heatmaps are created. Finally, predictions are fitted to a kinematic skeleton model.

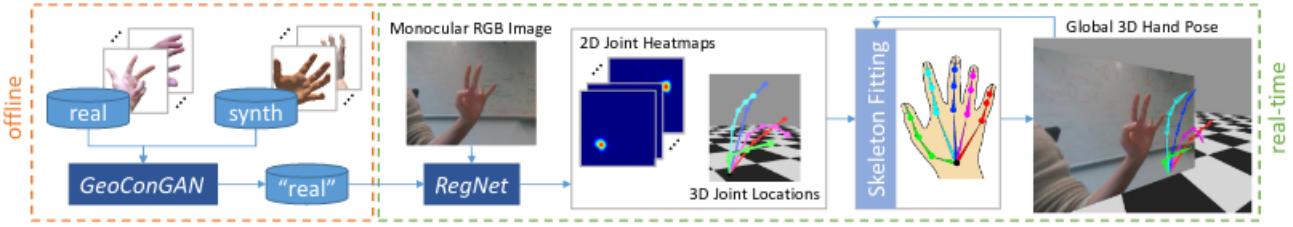


FIGURE 4.3: Network pipeline [37]

One of the most recent and competitive methods in this task [17] proposes a hand-model regularized graph refinement network under an adversarial learning framework. The network consists of three modules, a generator, a GCN refinement module and a discriminator. The generator aims to generate an initial estimation of 3D hand pose P, which serves as a prior pose for the hand pose refinement. Specifically, the hand region from the input image is fed into the ResNet-50 to extract features, which will be used to estimate the parameters of the MANO hand model. The MANO hand model is a deformable hand mesh model with two vectors θ and β, which control the pose and

shape of the hand. Consequently, the pose is projected to the 3D space via parameters that model the camera coordinate system, the 3D rotation parameter, the 3D rotation parameter and the scale parameter.

The objective of the GCN refinement module is to refine the structure of the prior pose. A Graph Res-block is designed to learn the deformation of the prior pose. Specifically, features are extracted from the RGB image via a ResNet-50 [16] architecture. In order to extract 2D features from the image Graph Res-block are used, which essentially aggregate information across adjacent nodes to extract higher-level features. Each Graph Res-block consists of two GCN layers as well as two normalization layers.

A multi-source discriminator is designed. The inputs include: 1) features of the input monocular image; 2) features of the refined hand pose; 3) features of bones computed from the refined hand pose. The bone features contain structural information such as the length and direction of bones. Specifically, a CNN is used to extract the features of the input monocular image, a GCN learns the representation of the refined hand pose and a fully-connected layer captures bone features computed from the refined hand pose.

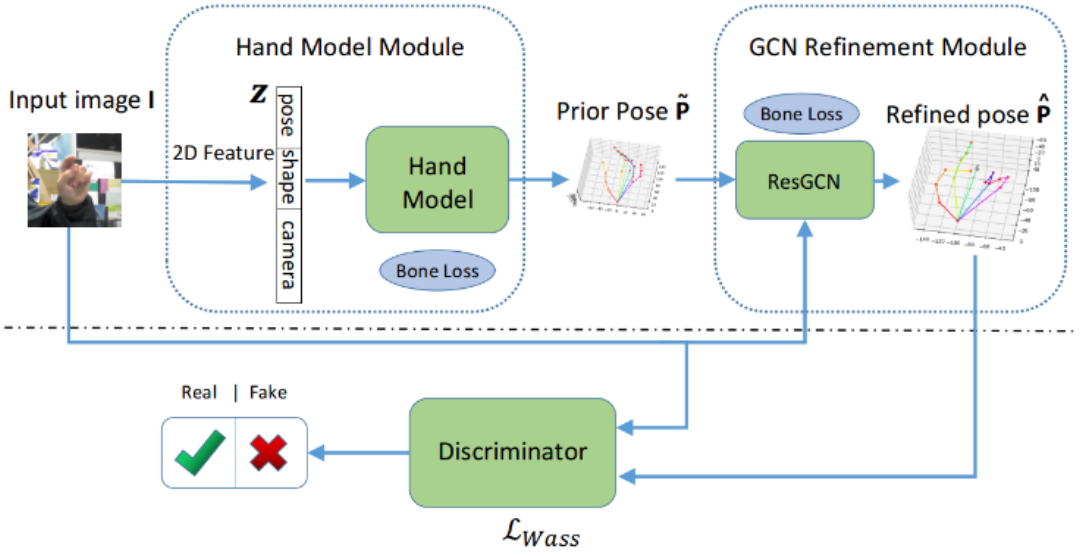


FIGURE 4.4: Graph Refinement Network pipeline [17].

4.2 Previous works on Sign Language Recognition

In this section we present some recent successful methods, that demonstrate competitive results on the problem of sign language recognition.

4.2.1 Spatial-Temporal Multi-Cue Network

Exploiting the synergy between multiple visual cues in sign language recognition presents many challenges. [60] proposes a framework that consists of three modules.

The SMC module extracts the spatial representation of each video frame. Using a monocular RGB frame as input it generates full-frame, hand, pose and mouth features. Specifically, a VGG-11 model is selected as a backbone network and the crops of the face and hands from the feature maps are fed to a series of convolutional layers. Their outputs are concatenated along the channel dimension. At the same time two deconvolutional layers are added after the 7th VGG-11 convolutional layer, whose output is processed by a point-wise convolutional layer to generate heat maps of each keypoint. The SMC block architecture is outlined below.

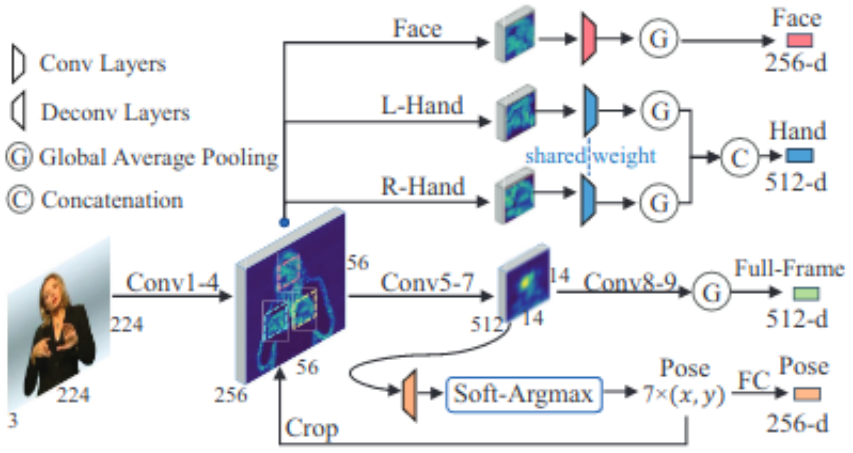


FIGURE 4.5: SMC Module Architecture [60].

The temporal multi-cue (TMC) module integrates information from two paths, intra-cue and inter-cue. The intra-cue path learns features from every cue seperately and at different time scales. Inside each cue temporal modeling is performed as:

$$f_{l,n} = ReLU(K_k^{\frac{C}{N}}(f_{l-1,n}))$$

$$f_l = [f_{l,1}, f_{l,2}, ..., f_{l,N}]$$

where $f_{l,n}$ represents the feature matrix of n-th cue and $K_k^{\frac{C}{N}}$ represents the kernel of temporal convolution. The inter-cue path temporally transforms the inter-cue feature from the previous block and fuses information from the intra-cue path.

$$o_l = ReLU([K_k^{\frac{C}{2}}(o_{l-1}), K_1^{\frac{C}{2}}(f_l)])$$

After each block a temporal max pooling is performed. The architecture of the TMC block is outlined below.

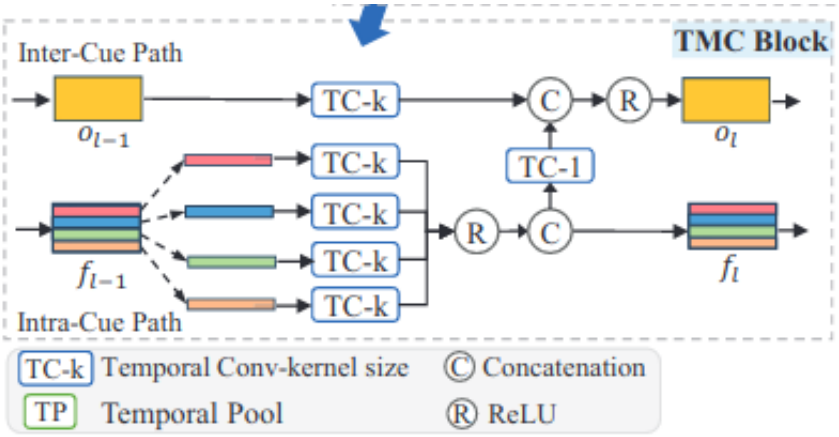


FIGURE 4.6: TMC Module Architecture [60].

Finally the two generated spatial-temporal feature sequences are used as input to a Bi-LSTM network, which models state transitions to achieve sequence learning.

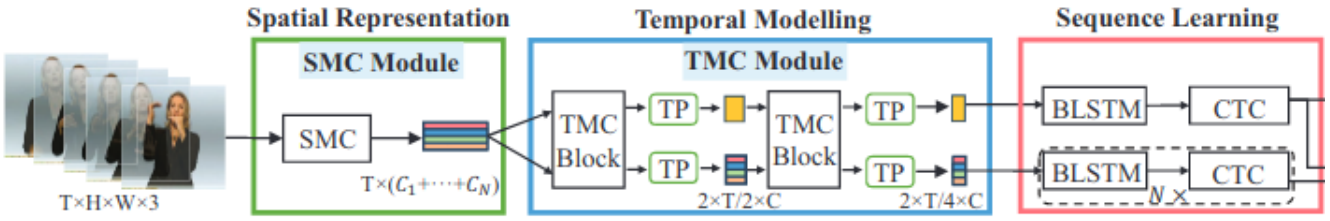


FIGURE 4.7: STMC Network Architecture [60].

4.2.2 Multi-Stream CNN-LSTM-HMMs

The proposed approach [29] embeds CNN-LSTM classifiers in multi-stream HMMs and adds synchronisation points between the different streams.

If we examine each stream separately the proposed algorithm performs the following steps. First, a random weak label sequence is predicted and the video is linearly segmented according to the sequence length. This frame labelling is used to train a CNN-LSTM network, which is the maximization step. Afterwards, the model is used in a CNN-LSTM-HMM setup, which refines the frame alignment. The HMM emission probabilities of the symbols are estimated by the CNN-LSTM. This constitutes the expectation step. The algorithm performs multiple expectation and maximization iterations until convergence.

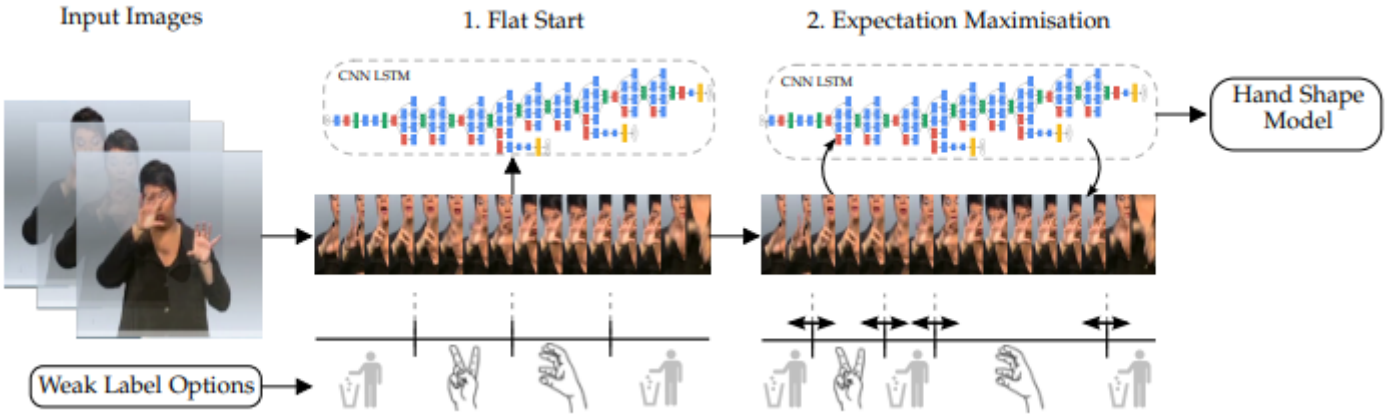


FIGURE 4.8: Single CNN-HMM Stream [29]

The problem of sign language recognition can be split into parallel sub-problems. Both manual parameters (hand shape, orientation, articulation and movement) and non-manual parameters (mouth shapes, facial expression and body orientation) occur in parallel to represent each sign. In order to extend this approach to include multiple streams intermediate synchronization points are incorporated. The expectation step is modified and synchronisation constraints are added in the HMM. they recombine the posterior of all independent streams into a single posterior probability via weighted summation. The approach is depicted below:

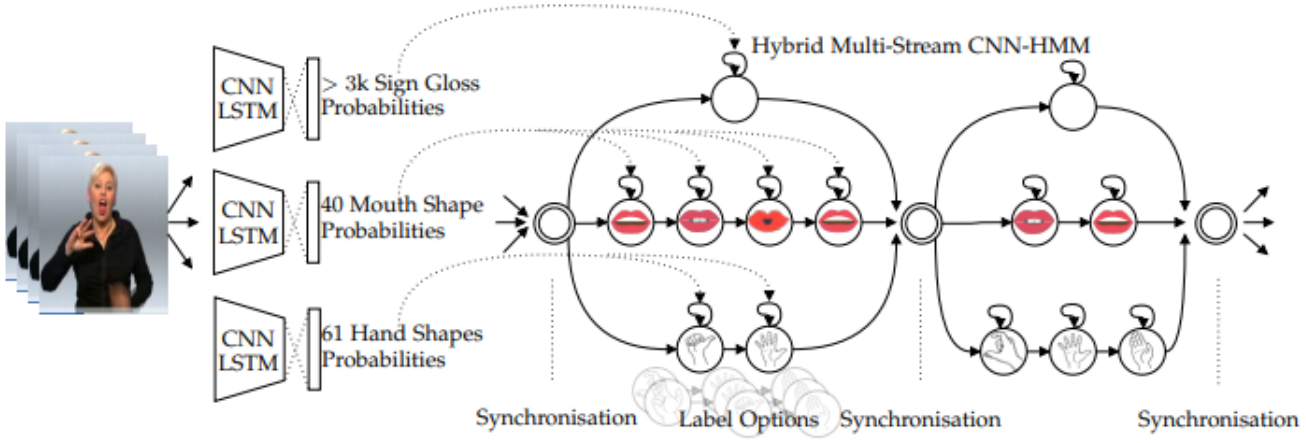


FIGURE 4.9: Multi-stream CNN-HMM [29].

4.2.3 Dilated 3D Convolutional Network

[45] leverages the power of 3D Residual Convolutional Networks (3D-ResNets) for visual feature extraction and uses stacked dilated convolutional layers for sequence modeling. Namely, in this work an 18-layer 3D-ResNet is used, which replaces 2D convolutions with

3D convolutions. 3D convolutions encode relationships in the 3D space and thus leverage temporal connections across frames. The architecture of the 3D ResNet is outlined below.

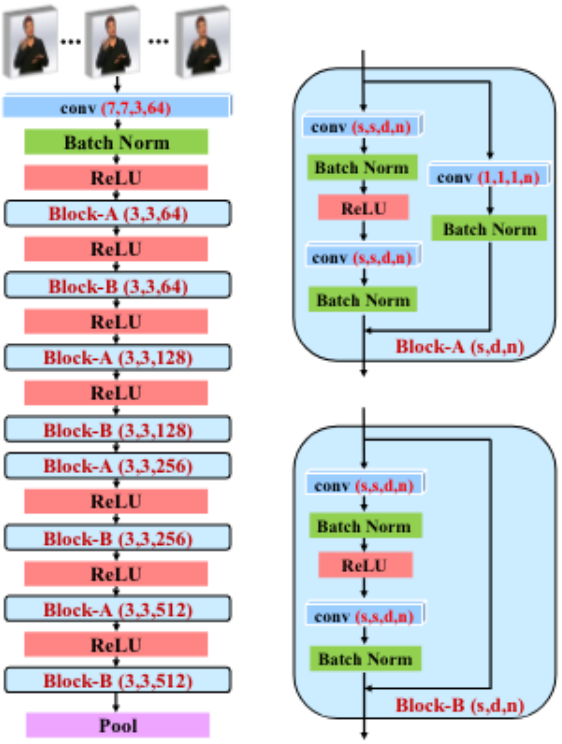


FIGURE 4.10: 3D ResNet architecture [45]

Dilated convolutions enable the network to learn temporal dependencies with varying resolutions for the input sequences, since different dilations have different receptive fields. Formally the ouputs o_i and h_i for the ith dilation cell are :

$$z = tanh(C_d(h_t^{(i-1)})) \odot \sigma(C_d(h_t^{(i-1)}))$$

$$o_t^i = tanh(C_{1*1}(z))$$

$$h_t^i = h_t^{i-1} + o_t$$

where C_d and C_{1*1} denote dilated convolution and 1x1 convolution

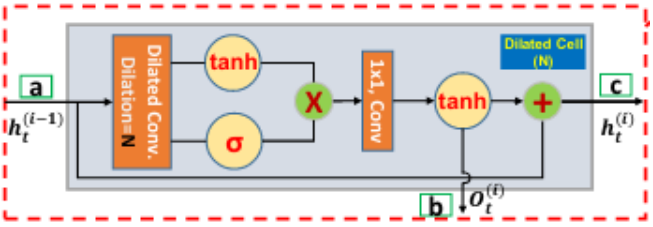


FIGURE 4.11: Architecture of the dilated convolutional cell [45].

The complete network architecture is demonstrated below. The outputs of the dilated convolutional module is the sum of the dilated cell outputs.

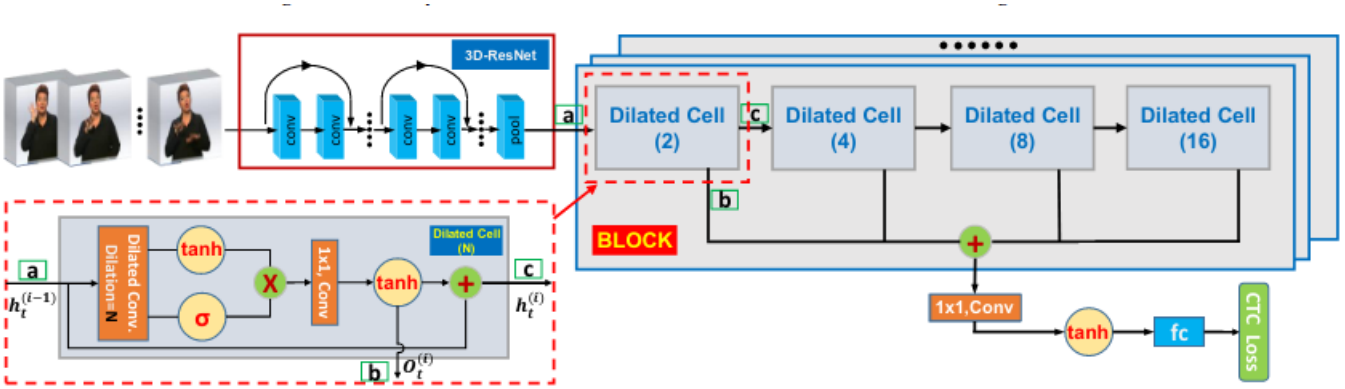


FIGURE 4.12: Network architecture [45].

Chapter 5

Experiments on Skeleton-based Isolated Sign Language Recognition

In this chapter we describe our preliminary experiments on Isolated Sign Language Recognition. We experiment with skeleton-based approaches from action and gesture recognition and report our findings.

5.1 Temporal Convolutional Networks

Due to the recent successful use of Temporal Convolution architectures [3] in action and gesture recognition we can apply corresponding architectures to the recognition of sign language. TCNs are used for sequence modeling tasks. They employ 1D convolutional operations and operate on the elements from current timestamp or earlier in the previous layer. These networks are characterized by layers in which convolution is performed with a time window d_l using N_l filters. Each filter examines d_l time steps across all feature dimensions. Therefore, it has a large activation when the input presents a sequence whose structure is related to that of the filter.

We start our experiments with a baseline architecture that uses residual blocks, in which 2 layers of dilated convolution, weight normalization, non-linearity and dropout are applied. We choose to connect three blocks with dilation factors 1,2 and 4 respectively and a convolution time constant equal to 3. To obtain a prediction we introduce a Global Average Pooling layer along the time domain and a softmax layer. The following figure demonstrates the architecture of a block, as proposed in [3]. The results are recorded in Table 5.1. It should be noted here that network training requires significantly less time and has a smaller memory footprint than recurrent architectures.

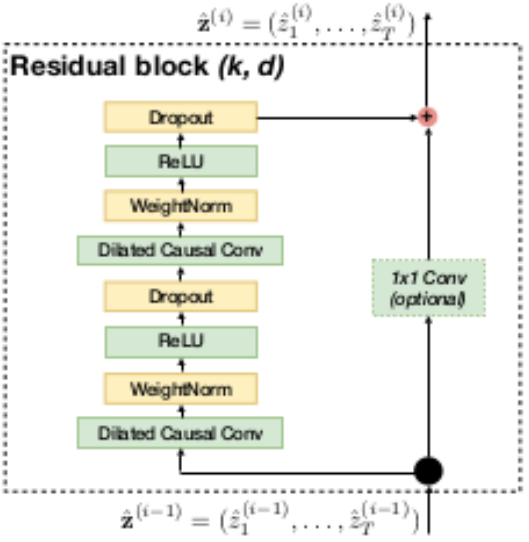


FIGURE 5.1: Block Architecture [3] .

In the second stage we apply the architecture which is presented in [25] and is also shown in the figure below. We observe similar results in performance. Given a Res-Net with N residual units the final representation results as:

$$X_N = X_1 + \ sum_{i=2}^{N} W_i * max(0, X_{i-1})$$

An additional advantage of this method is the ability to interpret the weights of convolutional filters. For example in filters of the first level W_1 each dimension corresponds to the coordinate of a joint at time t.

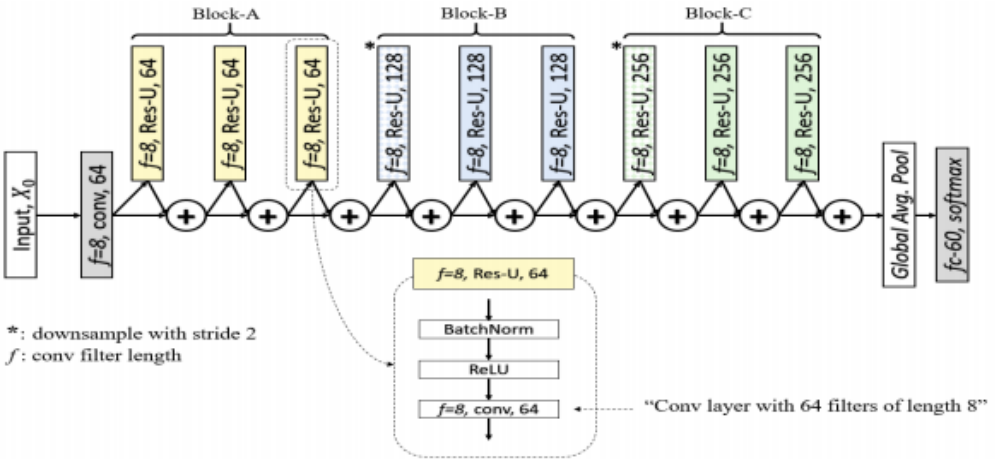


FIGURE 5.2: Res-TCN Model Architecture [25].

However, success in sign recognition also depends on the model's ability to distinguish between handshapes. Due to the fact that gestures present high inter-class variance, not all the features extracted from the TCN provide useful information. Therefore, an effective solution is to add a attention mechanism, which will help the model to focus on specific information derived from frames and features. For this purpose we use the following network [19], in which a main branch is complemented by a parallel branch that calculates attention.

Specifically, given the input of the previous block $X_{l-1}\epsilon\mathbb{R}^{T\times N_{l-1}}$ from the main branch and $M_{l-1}\epsilon\mathbb{R}^{T\times N_{l-1}}$ from the mask branch the new feature maps generated are computed as follows:

$$\tilde{X}_l = X_{l-1} + F(W_{lmain}, X_{l-1})$$

$$\tilde{M}_l = G(W_{lmain}, \tilde{M_{l-1}})$$

where W_{lmain} and W_{lmask} are temporal filters and F() and G() denote batch normalization, dropout and ReLU activation.

$\tilde{M}_l^i = \tilde{m}_{l,1}^i, ..., \tilde{m}_{l,T}^i$ are scores indicating the importance of each time frame. $\tilde{M}_{l,t}^i = \tilde{m}_{l,t}^1, ..., \tilde{m}_{l,t}^N$ are scores indicating the importance of channel. By performing element-by-element multiplication between the characteristics of the main loop X_l and the mask M_l we obtain characteristics with "spatio-temporal awareness". The model shows significantly improved performance from the previous ones.

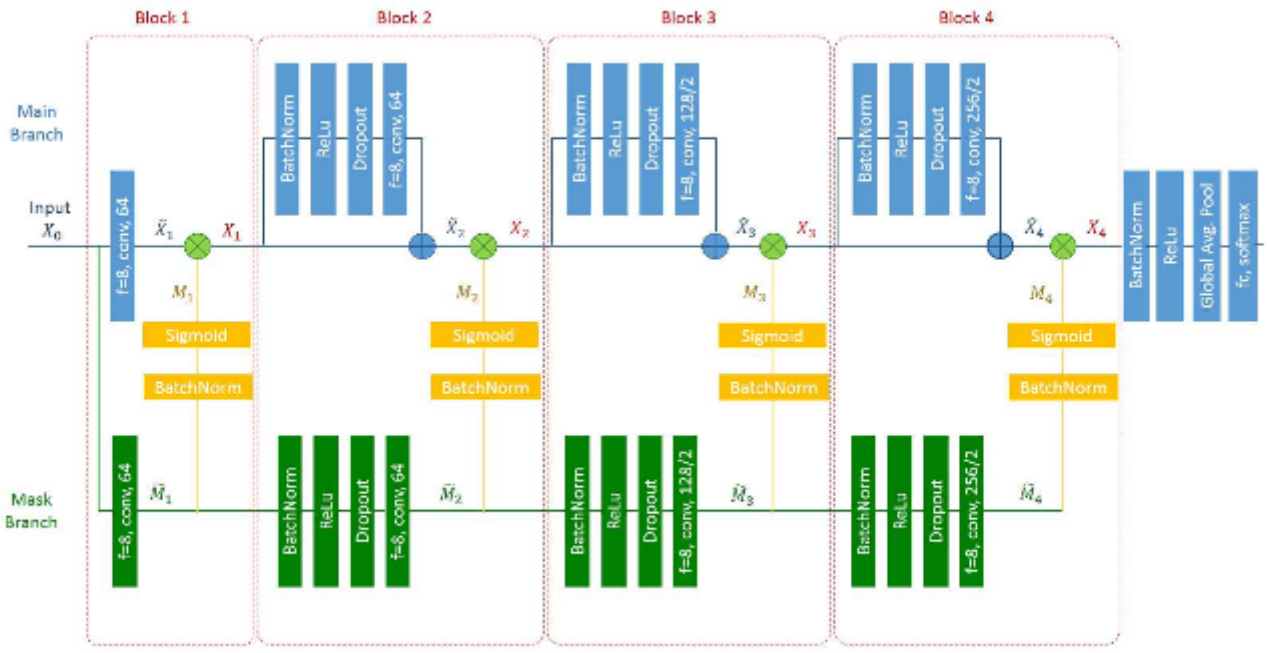


FIGURE 5.3: STA-ResNet Model Architecture [19]

5.2 Hierarchical CNN

After reviewing methods that employ Temporal Convolutional Networks, we proceed with CNN based models. Our main method is the one proposed in [34] , which allows us to get the overall response from all the joints and thus take advantage of the correlations between them. It draws on the fact that in convolutional networks information from a dimension can be aggregated globally if is specified as channels while the other two encode local context.

Our input is modeled as an image in which the y axis represents the frames and x axis the joints. The channels correspond to the dimension of the coordinates of the joints. Firstly, we extract pointwise characteristics for each joint separately, using 1×1 and $n \times 1$ kernels. In the second stage, by viewing the dimension of the joints as the channel dimension, the next convolutional layers can hierarchically extract features from all the joints and in this way model interactions between different joints. The results of the method are presented below. Note here that the network takes as input both the joint positions and the movement of the skeleton, which is defined as the difference between the

position of successive joints. This addition helps the model learn the temporal evolution pattern of the input.

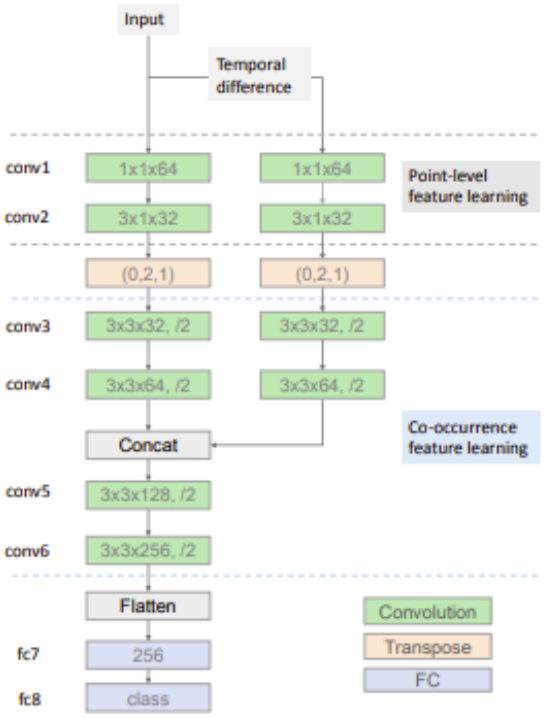


FIGURE 5.4: HCN model Architecture [34].

5.3 Experiments on GSLL dataset

We test the effectiveness of the above methods on the Greek Sign Language Lemmas Dataset (GSLL) [54]. It consists of two signers and has a vocabulary of 1043 different glosses. We limit our experiments to a subset, containing 300 glosses. Each gloss is performed five times by each signer and thus the total number of videos is 3000.

Method	Top 1 Accuracy	Top 3 Accuracy
3-layer BiLSTM	80 %	95 %
Res-TCN (1st approach)	90 %	97 %
Res-TCN (2nd approach)	87 %	97 %
Res-TCN-STA	95 %	99 %
Hierarchical CNN	86 %	96 %

TABLE 5.1: Performance comparison of different skeleton-based methods on GSLL dataset.

Chapter 6

3D Hand and Body Pose Estimation

In this chapter we analyze our method on 3D hand and body pose estimation and conduct experiments with alternative architectures. We also provide qualitative in-the-wild examples and extensive quantitative results on three popular fully annotated pose corpora.

6.1 3D hand pose lifting approach

Our approach [45] extracts 3D hand-joint keypoints by "lifting" 2D joint locations to the 3D space. Our input is a vector of 2D hand-joint keypoints, generated by the OpenPose framework, and our output is a series of points in the 3D space. We zero-center both 2D and 3D poses around the wrist joint, so as to ensure that our model learns translation-invariant representations. A significant source of error in 3D joint predictions is the existence of noise in the input 2D predictions. Thus, we apply smoothing to the input via a median filter with radius one to remove noise spikes and eliminate unstable predictions.

$$\tilde{\alpha}[n] = m(\alpha[n-T], ..., \alpha[n], ...\alpha[n+T])$$

,where $\tilde{a}$ denotes the preprocessed signal, $m()$ is the median operator and 2T+1 is the size of the filter.

Consequently, we implement a simple but powerful architecture, originally proposed in [36] for human pose 3D estimation. Our model is a deep multi-layer neural network containing batch normalization, dropout, rectified linear units (ReLU), and residual connections. The latter boosts generalization performance, while batch normalization and dropout boost model robustness to noisy detections. Moreover, we apply a constraint on the weights of each layer, so that the maximum norm is less than or equal to one.

Specifically, the building block of the network is a linear layer followed by batch normalization [20], dropout [52], and ReLU activation. This block is repeated twice, and the two blocks share a residual connection. For this task we stack two outer residual blocks and our model contains approximately 4 million trainable parameters.

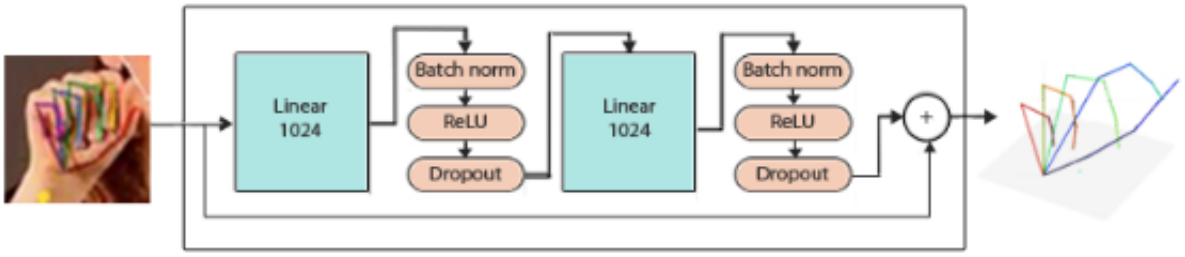


FIGURE 6.1: The main building blocks of the 3D Pose estimation network.

For network training, we use the Rendered HandPose Dataset [61], a large-scale 3D hand pose dataset based on synthetic hand models. The model yields 21 3D joints for each hand. The wrist is assumed as the coordinate system origin.

6.2 Experiments

We evaluate our 3D hand pose generation network performance using two corpora: the Rendered HandPose dataset (RHD) and the FreiHand database (FHD) [62].

Rendered HandPose dataset : We use this corpus for network training. It constitutes a large-scale 3D hand pose dataset, based on synthetic hand models. The dataset utilizes 3D human models with corresponding animations from Mixamo 2. It features 20 different characters performing 39 actions, and for each frame a different camera location is randomly selected. The train set contains 41,258 images and the evaluation set 2,728 images with a resolution of 320×320 pixels. Annotations of a 21 keypoint skeleton model of each hand are provided, as well as segmentation masks. In our work, we utilize the 2D hand joint coordinates in the image frame and the 3D joint positions in the world frame.

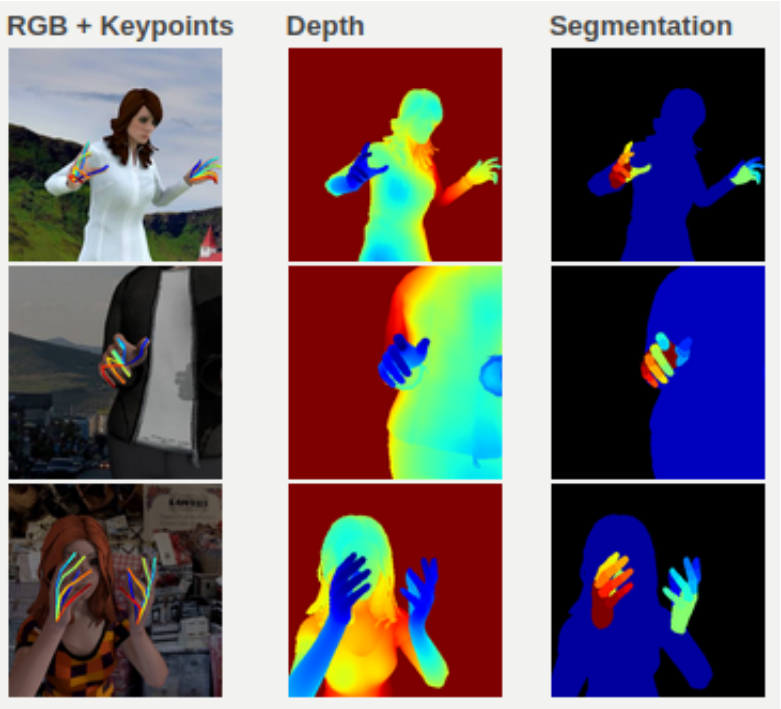


FIGURE 6.2: Example images from RHD dataset [61].

FreiHAND Dataset : The dataset consists of real images and provides samples both with and without object interactions. It is captured with a multi-view setup and contains 33,000 samples. Hand poses are recorded from 32 subjects, and the set of actions feature ASL signs, counting and moving fingers to their kinematic limits. 3D annotations for 21 hand keypoints are also provided. For this work, the data is partitioned to 80% for training and 20% for testing.

FIGURE 6.3: Example images from FreiHAND dataset [62].

In the following table and figure, we evaluate the performance and generalization power of our model on the aforementioned datasets for various training / testing setups. We report average median point error per keypoint of the predicted 3D pose, when given the 2D ground truth pose, as well as the area under the curve (AUC) on the percentage of correct keypoints for different error thresholds. In order to measure the cross-dataset generalization of our network, we use the model trained on RHD-train and report AUC score and median error per joint on the FHD dataset, after alignment with the ground truth (Procrustes analysis). We also report the percentage of correct key-points (PCK), which is defined as the mean percentage of predicted joints below an Euclidean distance from the correct joint location. According to our results, our method demonstrates competitive performance on both datasets.

The RHD set can be characterized as more challenging, due to its variations in viewpoints, and as a result we report higher 3D pose error. Since we are mostly interested in the generalization ability of our model and its performance "in the wild", we observe that our model can adapt effectively to unseen data and accurately localize the hand pose.

Training	Testing	AUC score	Median error per joint
RHD-Train	RHD-train	0.729	18.1
	RHD-test	0.616	22.6
	FHD-test	0.771	16.2
FHD-Train	FHD-test	0.900	11.0

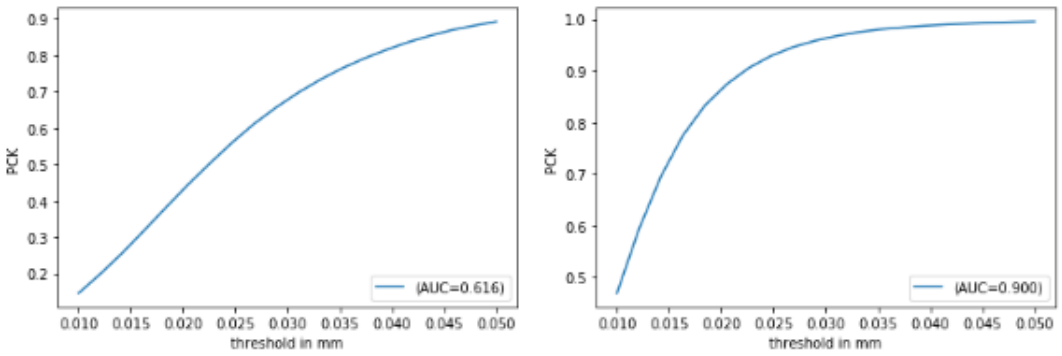


FIGURE 6.4: Percentage of correct keypoints (PCK) over a certain threshold in mm, evaluated: (a) on RHD-test for model trained on RHD-train and (b) on FHD-test for model trained on FHD-train.

We also test our method qualitatively in order to determine whether it can generalize to unseen videos "in the wild". The following examples are 3D hand skeletons produced by our model on unseen sign language videos.

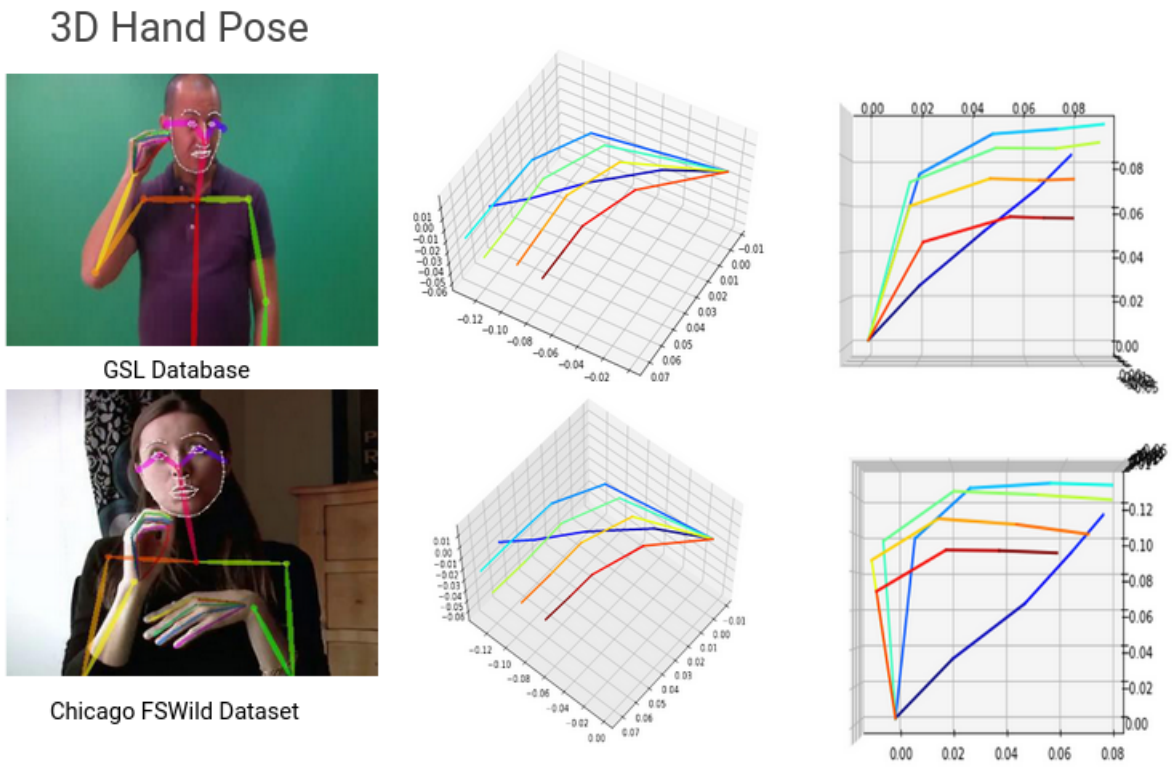


FIGURE 6.5: 3D Hand Pose estimation examples from the GSLL [54] database and the Chicago FSWild Dataset.

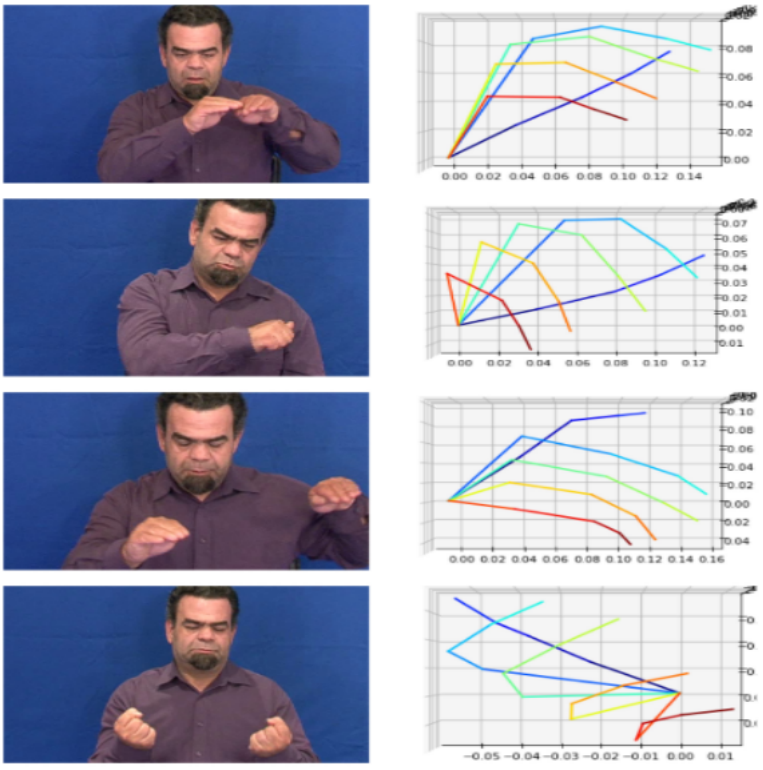


FIGURE 6.6: 3D Hand Pose estimation examples from the GSL dataset.

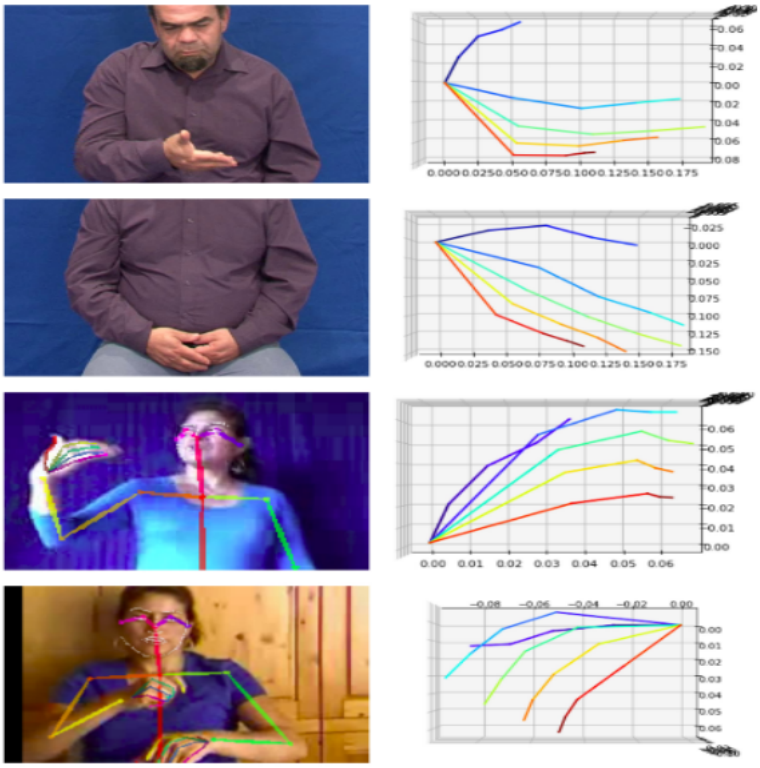


FIGURE 6.7: 3D Hand Pose estimation examples from the GSL dataset and ChaLearn.

Small losses in the 3D euclidean space do not always translate to small distortions. Drawing on the method, introduced in [17], which estimates 3D hand pose via hand-model regularized graph refinement under an adversarial learning framework, we employed two additional custom loss functions and incorporated them to our model: a bone length loss function L_{len}, which estimates the difference between ground truth bone length and its prediction and a bone direction loss L_{dir}, which measures deviation in the direction of bones. Formally,

$$L_{len} = \sum_{i=1}^{N} \sum_{j=1}^{N} \left| \|b_{ij}\| - \|\hat{b_{ij}}\| \right|$$

$$L_{dir} = \sum_{i=1}^{N} \sum_{j=1}^{N} \left\| \frac{b_{ij}}{\|b_{ij}\|} - \frac{\hat{b_{ij}}}{\|\hat{b_{ij}}\|} \right\|$$

where $b_{ij} = j_i - j_j$ denotes the bone vector between joints i and j

The resulting loss function is their weighted sum:

$$L_{loss} = L_{joint} + \lambda_{len} L_{len} + \lambda_{dir} L_{dir}$$

where L_{joint} represents the 3D Euclidean distance between joints.

6.3 3D Body Pose Estimation

Hand articulation plays a vital role in sign language recognition. However, knowledge about the trajectory of the arm joints in the three dimensional pane can also provide necessary information and aid the model in distinguishing between signs with similar handshape.

An identical architecture can be exploited to "lift" body keypoints to the 3D space, as proposed in the original paper. We retrain the network for 200 epochs, using a subset of the Human 3.6 M dataset. Our primary task is sign language recognition and in this setting usually only upper body poses are visible. Thus, we set the occluded keypoints to 0 during training, so that our model has the ability to capture poses, where the lower body is not visible. We report a 20.3 mean error per joint.

Human 3.6M: Human 3.6M [9, 21] is a 3D human pose dataset containing 3.6 million human poses and corresponding images. For each frame body joint positions in the 3D space as well as their 2D projections are provided, as well as camera parameters and body proportions. The dataset features 11 different actors in 17 scenarios. High-resolution 50Hz video is captured from 4 calibrated cameras. For training purposes we use a subset of the dataset.

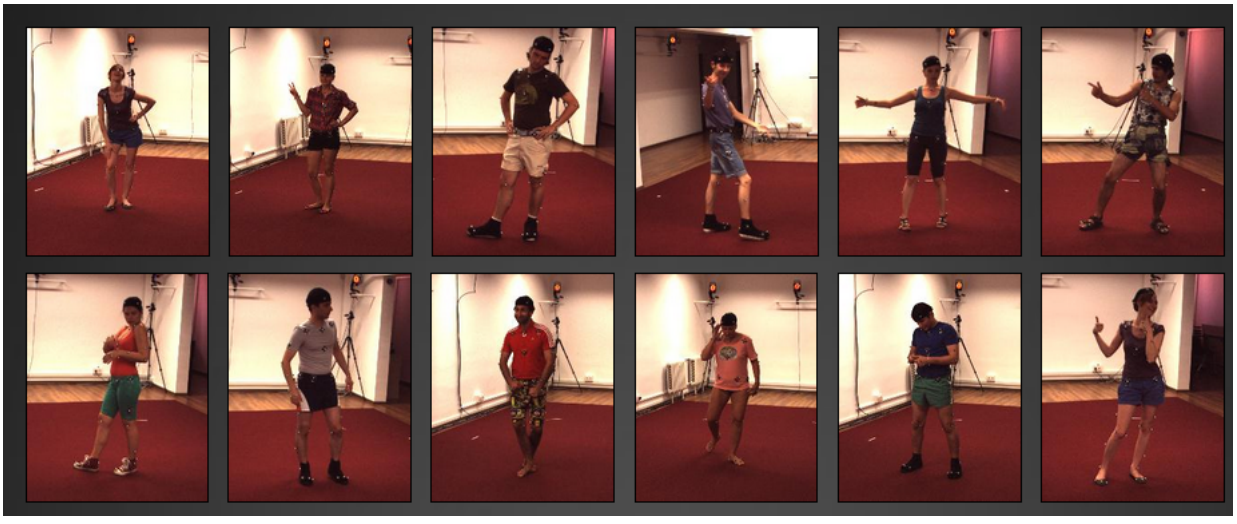


FIGURE 6.8: Example frames from Human 3.6 dataset featuring different actors [21]

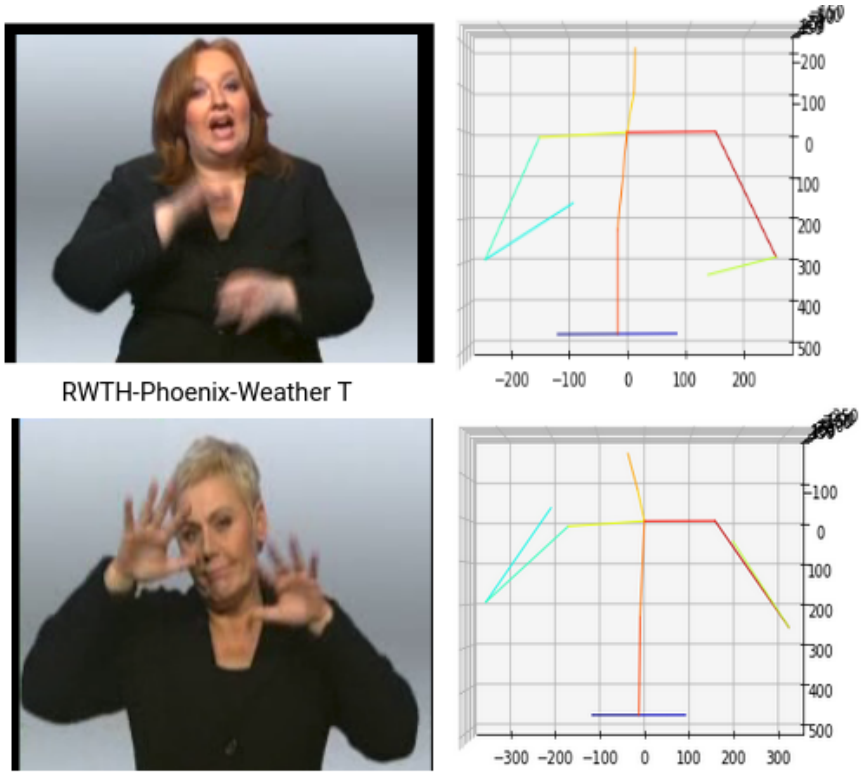


FIGURE 6.9: 3D Body Pose estimation examples from the RWTH Phoenix 2014T dataset

6.4 Semantic Graph Convolutional Networks for 3D Hand Pose Estimation

Inspired by the success of graph convolutional networks for input, that follows a graph structure we can exploit graph convolutions to provide a solution to the problem of 3D hand pose regression. Both 2D and 3D hand pose can be represented as a structured graph, encoding the relationship between hand joints. Following the work of [59], we utilize an enhanced graph convolution operation called Semantic Graph Convolution, which learns channel-wise weights for the edges. This operation is combined with non-local layers [57] to ensure that both local and global dependencies among nodes are captured.

Semantic graph convolutions approximate CNNs, which learn a separate transformation matrix for each feature in the kernel by learning a weighting vector $\vec{a_i}$ for each position and consequently aggregating them via a shared transformation matrix W. Semantic graph convolution operation can be formulated as:

$$X^{l+1} = \sigma(W X^l \rho_i(M \odot A))$$

where ρ_i is the softmax nonlinearity which normalizes the input matrix and $M \epsilon \mathbb{R}^{K \times K}$ is a learnable mask.

We can extend this equation by learning a set $M_d \epsilon \mathbb{R}^{K \times K}$ so that each channel d of the output undergoes a different weighting matrix.

$$X^{l+1} = \Big\Vert_{d=1}^{D_{l+1}} \sigma(\vec{w}_d X^l \rho_i(M_d \odot A))$$

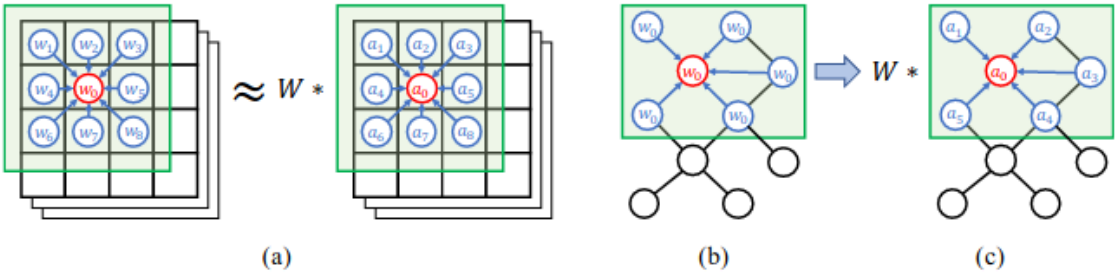


FIGURE 6.10: Ilustration of semantic graph convolution [59].

This extends ST-GCN, mainly by learning channel-wise different weights for edges instead of a single learnable mask for all channels.

In order to capture long range dependencies among nodes we enhance the feature updating mechanism to compute responses between distant nodes. Thus, we define a non-local mean operator as:

$$x_i^{(\vec{l+1})} = x_i^{(\vec{l})} + \frac{W_x}{K} \sum_{j=1} f(x_i^{(\vec{l})}, x_j^{(\vec{l})}) g(x_j^{(\vec{l})})$$

where f is a pairwise function to calculate the affinity between node i and all other j, g computes the representation of the node j.

6.4.1 Network Architecture

The main building module of the architecture is a Semantic Graph Convolutional layer with 128 channels followed by Batch Normalization and ReLU activation, which is repeated twice. Afterwards the output passes through a non-local layer. This block is repeated four times.

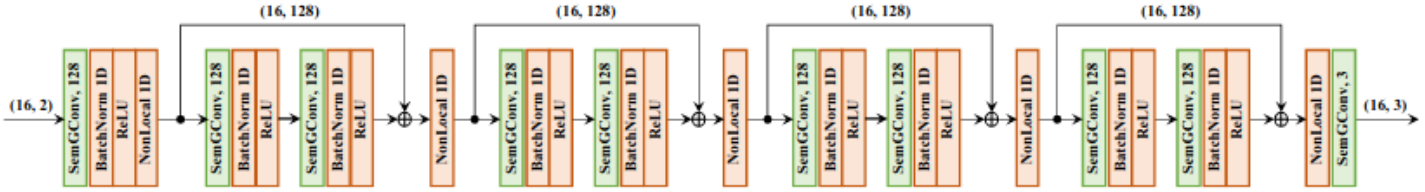


FIGURE 6.11: Architecture of semantic graph convolutional network [59].

We implement this method and use the Rendered Hand Pose Dataset (RHD) for training. However, the performance is inferior to the deep fully connected model, described above.

Chapter 7

Continuous Sign Language Recognition

In this chapter we describe our multi-stream approach on the task of continuous sign language recognition. We provide theoretical background on connectionist temporal classification loss and spatio-temporal graph convolutional networks. We also compare our approach to baseline methods and report and analyze our results.

7.1 Connectionist Temporal Classification

Before we present our approaches on Sign Language recognition we provide theoretical background on the loss function we will use for continuous sign language recognition.

Due to the lack of frame level annotations it is not possible to use cross entropy loss for training. However, an alternative form of weaker supervision is to use a sequence-to-sequence learning loss function, namely CTC that can be used for unsegmented data [14]. Given frame level gloss probabilities, $p(g_t|V)$ we can use CTC to compute $p(G|V)$ by marginalizing over all possible V to G alignments:

$$p(G|V) = \sum_{\pi \epsilon B} p(\pi|B)$$

where π is a path and B are the set paths that correspond to G.

B is defined as a many-to-one mapping operation. It firstly removes the repeated labels then removes all blanks from the given path. Given a label sequence l, we define feasible paths as all π that can be mapped on to g through B. CTC can be computed effectively through dynamic programming thanks to the Markov property.

Specifically,

$$a_t(s) = \sum_{B(\pi_{1:t})=l_{1:s}} \prod_{t=1}^{t} y_{\pi t}^t$$

where:

y_k^t : is the output at time t for symbol k

l is the label

l' is the label with blanks

The recurrence rule can be defined as:

$$a_1(1) = y_b^1$$

$$a_1(2) = y_{l1}^1$$

$$a_1(s) = 0 \; \forall s > 2$$

$$a_t(s) = \begin{cases} a_{\bar{t}}(s) y^t l_s & if l_s = b \; or \; l_{s-2} \\ a_{\bar{t}}(s) + a_{t-1}(s-2) y_{l_s}^t & otherwise \end{cases}$$

$$a_{\bar{t}}(s) + a_{t-1}(s) + a_{t-1}(s-1)$$

$$p(l|x) = a_T(|l|) + a_T(|l|-1)$$

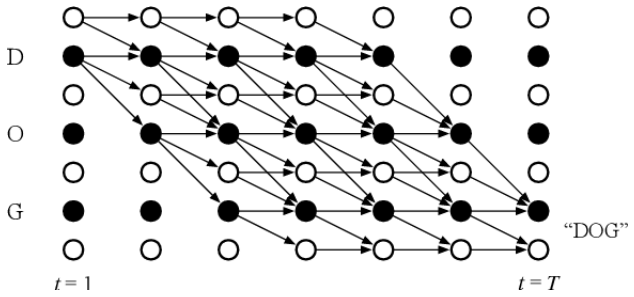


FIGURE 7.1: Forward algorithm of CTC for the target sequence "DOG".

7.2 Sign Language Transformer

As a strong baseline for our experiments we rely on the Transformer architecture, which has demonstrated very promising results in our task. Following [5] we initially embed

our source tokens, namely sign language video frames. We use the Spatial Embedding approach and pass the image through a CNN encoder. Additionally, since transformers do not employ recurrence or convolution we add "positional encodings" to our input embeddings in order to inject temporal information. Sine and cosine functions of different frequencies are used:

$$PE_{(pos,2_i)} = sin(pos/1000^{2i/d_{model}})$$

$$PE_{(pos,2_{i+1})} = cos(pos/1000^{2i/d_{model}})$$

,where *pos* is the position and i is the dimension.

Consequently, the input passes through three stacked transformer encoder layers. In each layer the input is first modelled by a self-attention layer which learns the contextual relationship between the frame representations of a video. Outputs of the self-attention are then passed through a non-linear point-wise feed forward layer. These operations are followed by residual connections and normalization. The extracted spatiotemporal representation of each frame is fed to a linear layer with softmax activation for prediction. Due to the absence of frame labels we employ Connectionist Temporal Classification Loss for training.

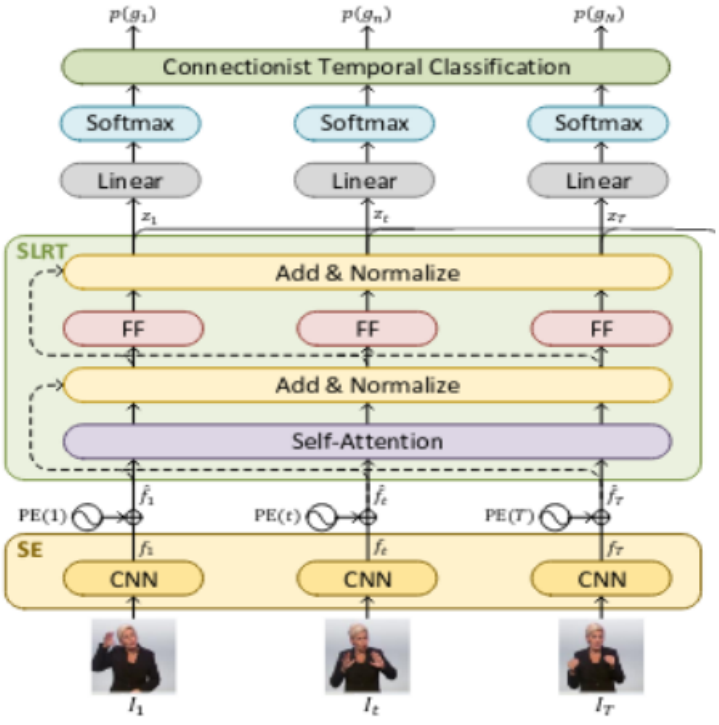


FIGURE 7.2: Sign Language Transformer architecture [5].

Drawing on [43] we add a regularization term to the CTC objective function which consists of the Kullback-Leibler divergence [22] between the network's predicted distribution P and a uniform distribution U over all labels. Formally,

$$L(\theta) = (1-\alpha)L_{ctc} + \alpha \sum_{t=1}^{T} D_{KL}(P_t||U)$$

In this set of experiments we set $\alpha = 0.05$

This term penalizes low-entropy distributions and enhances the generalization ability of our model. In general label smoothing is a way to improve generalization by adding label noise, which has the effect of penalizing overly confident predictions.

7.2.1 Hybrid CTC/Attention

We can extend the previous approach by a multiobjective learning method which combines a CTC objective and an attention-based encoder network [24]. The attention-based approach estimates the posterior $p(C|X)$ as:

$$p_{att}(C|X) = \prod_{l=1}^{L} p(c_l|c_1, ..., c_{l-1}, X)$$

which can be obtained by

$h_t = Encoder(X)$

$a_{lt} = LocationAttention(a_{l-1}{}_{t=1}^{T}, q_{l-1}, h_t)$

$r_l = \sum_{t=1}^{T} a_{lt} h_t$

$p(c_l|c_{l-1}...c_1|X) = Decoder(r_l, q_{l-1}, c_{l-1})$

where a_{lt} is an attention weight, h_t is the representation for frame t extracted by the encoder.

In our experiments our encoder is a Transformer with 3 layers. Our decoder is a unidirectional 1-layer LSTM. The decoder outputs the hidden state q_t and accepts the hidden vector q_{l-1} and as input the concatenated vector of the label-wise hidden vector ,r_l, and the embedding of the previous output ,c_l,.

$$q_l = LSTM_l(r_l, q_{l-1}, c_{l-1})$$

Attention Mechanism

We conduct experiments with two different attention mechanisms. The location-aware attention mechanism extends content-based attention to include convolution. Specifically, the content based attention generates the attention distribution by calculating the similarity between the current hidden state of the decoder and the frame representation feature map. Intuitively we find the most correlated feature vectors in the feature map of the encoder, which can be exploited to predict the label at the current time step. In the next step the attention distribution is used to produce a weighted sum of the encoder hidden states, namely the context vector. Location aware attention addresses this limitation by explicitly taking into account the location information. Specifically, we extract k vectors f_t, for every position of the previous alignment a_{l-1} by convolving it with a matrix K:

$$f_{tt=1}^{T} = K * a_{l-1}$$

$$e_{lt} = g^T tanh(Lin(q_{l-1}) + Lin(h_t) + LinB(f_t))$$

$$a_{lt} = Softmax(e_{ltt=1}^{T})$$

where g is a learnable vector parameter, e_{lt} is a T-dimensional vector, * denotes convolution along the input feature axis,t, with the convolution parameter ,K, to produce the set of T features f_t.

We also conduct experiments with a different type of attention mechanism, namely a combination of coverage [48] and location aware attention [11]. The coverage mechanism addresses the problem of repetition. We maintain a coverage vector c_t, which is the sum of attention distributions over all previous decoder timesteps: $c_t = \sum_{t=1}^{T} a_t$. Intuitively, c_t represents the degree of coverage that those words have received from the attention mechanism so far. The coverage vector is used as an additional input to the attention mechanism. Formally,

$$e_{lt} = g^T tanh(Lin(q_{l-1}) + Lin(h_t) + LinB(f_t) + Lin(c_t))$$

This enables the attention mechanism to be informed through a summary of its previous decisions and eschew repeatedly attending to the same locations.

The encoder-decoder architecture is outlined below:

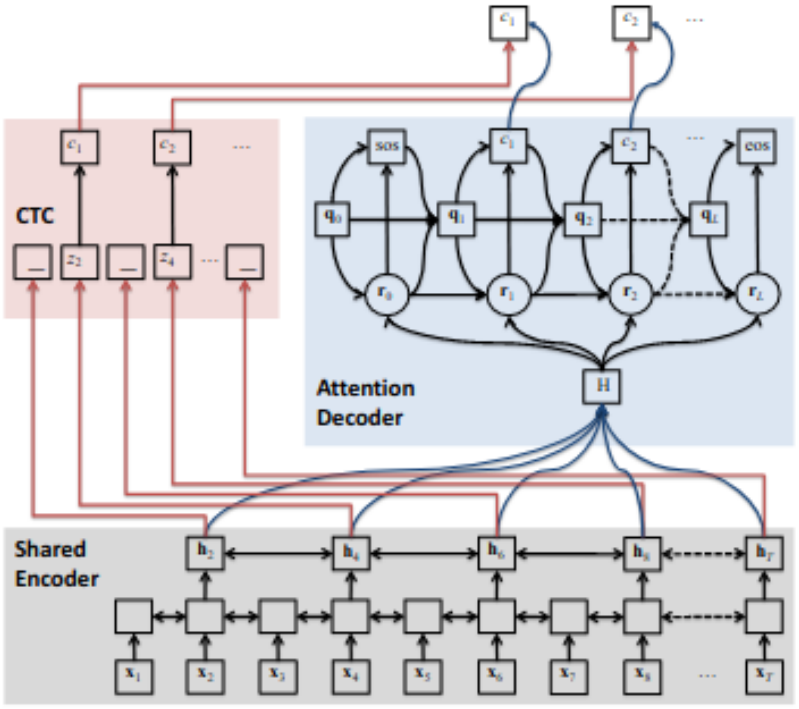


FIGURE 7.3: Hybrid CTC/Attention Encoder-Decoder Architecture. [58].

As previously stated we employ a multiobjective learning strategy where the objective is a weighted logarithmic linear combination of the CTC and attention objective.

$$L_{mol} = \lambda \log p_{ctc}(C|X) + (1 - \lambda) \log p_{att}(C|X)$$

We observe that $\lambda = 0.2$ yields the best results.

The inference step is performed by a label synchronous beam search decoding, that takes into consideration the CTC probabilities. The inference objective is defined as :

$$\hat{C} = \arg\max_{h \epsilon \hat{\Omega}} \{\lambda \alpha_{ctc}(h, X) + (1 - \lambda)\alpha_{att}(h, X)\}$$

where,

$\alpha_{ctc}(h, X) = \log p_{ctc}(h, X)$

$\alpha_{att}(h, X) = \log p_{att}(h, X)$

The results of these implemented methods are used as a strong baseline and are presented in conjuction with our method, which will be analyzed in the next sections.

7.3 Spatial-Temporal Graph Convolutional Networks

Our method is inspired by [50] that applies GCNs in the task of skeleton-based action recognition and proposes a Spatial Temporal Graph Convolutional Network (ST-GCN), which automatically extracts both the spatial and temporal patterns from data. The connectivity of the joints is a significant aspect of the problem and it is useful to capture the patterns embedded in their spatial configuration.

The model proposed is formulated on top of a sequence of skeleton graphs, where each node corresponds to a joint and there are two types of edges, the spatial edges that follow the natural connectivity of the joints and the temporal edges, that connect the same joints across consecutive time steps.

Formally, an undirected graph $G = (V, E)$ is constructed where the node set $V = (u_{ti}|t = 1, ..., T, i = 1, ..., N)$ includes all the joints in a skeleton sequence. The edge set is composed of two subsets, the first subset depicts the intra-skeleton connection at each frame, denoted as $E_s = (u_{ti}, u_{tj}|(i, j) \in H)$ where H is the set of naturally connected human body joints. The second subset contains the inter-frame edges, denoted as $E_f = (u_{ti}u_{t+1,i})$. Edges in E_f for a particular joint represents its trajectory over time.

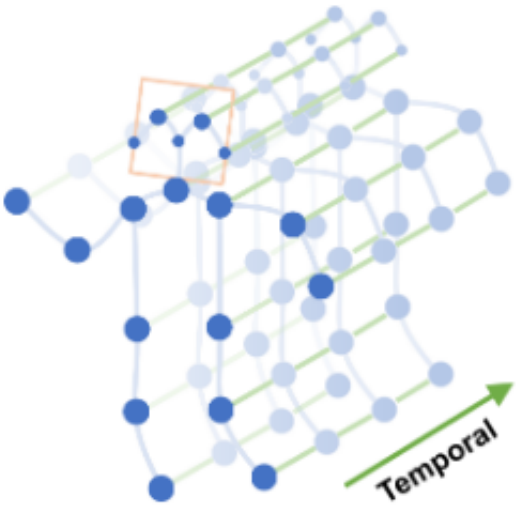


FIGURE 7.4: Spatial temporal graph of skeleton sequence [50].

Firstly, we present the spatial convolution operation on graphs, which is defined by extending the definition of convolution on 2D grids to cases where the features map resides on a spatial graph V_t. On graphs we define the neighbor set of a node as $B(u_{ti}) = u_{tj}|d(u_{tj}, u_{ti}) \leq D$. In this work $D = 1$ is used, that is, the 1 neighbor set of joint nodes.

In order to define the weight function we partition the neighboring set $B_{u_{ti}}$ into a fixed number of subsets, where each subset corresponds to a numeric label. Thus we have a mapping l_{ti} , which maps a node to its subset label. The weight function can be implemented by indexing a tensor of (c, K) dimension where K is the number of subsets. It is important to design an effective partitioning strategy to implement the label map. In this work a spatial configuration partitioning is employed. The neighbor set is divided into three subsets: 1) the root node; 2) the centripetal group: the neighboring nodes that are closer to the gravity center of the skeleton than the root node; 3) otherwise the centrifugal group.

Formally :

$$l_{ti}(u_{tj}) = \begin{cases} 0 & if\ r_j = r_i \\ 1 & if\ r_j < r_i \\ 2 & if\ r_j > r_i \end{cases}$$

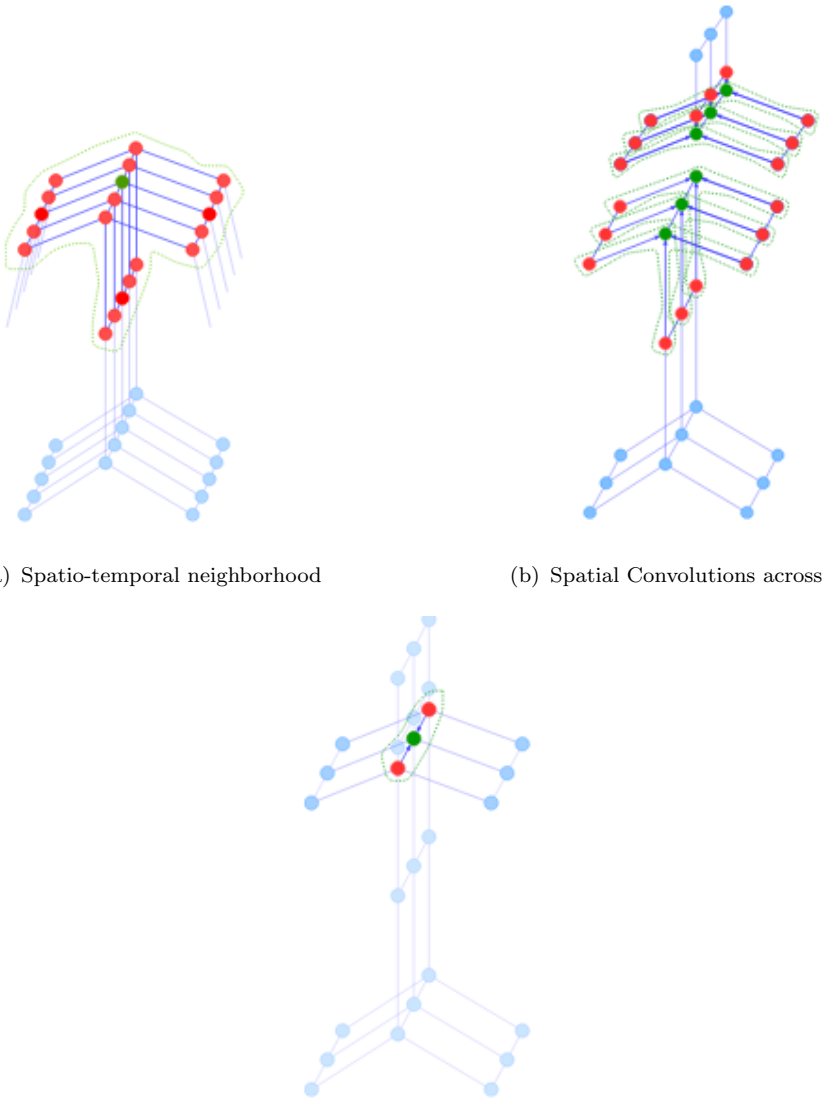
FIGURE 7.5: Spatial configuration partitioning.

7.3.1 Spatial and Temporal Graph Convolution

Spatial graph convolution is defined as:

$$f_{out}(u_{ti}) = \sum_{u_{tj} \in B_{u_{tij}}} \frac{1}{Z_{ti}(u_{tj})} f_{in}(u_{tj}) \cdot w(l_{ti}(u_{tj})$$

After the formulation of spatial graph convolution the next step is to extend the spatial graph CNN to the spatial temporal domain. Thus, we include temporally connected joints in the neighborhood as:



(a) Spatio-temporal neighborhood

(b) Spatial Convolutions across time

(c) Temporal neighborhood

FIGURE 7.6: Definition of neighborhoods [50]

$$B_{u_{ti}} = u_{qj} | d(u_{tj}, u_{ti}) \leq K, |q - t| \leq \lfloor \Gamma/2 \rfloor$$

where the parameter Γ controls the temporal range to be included in the neighbor graph. The label map l_{ST} is modified as follows:

$$l_{ST}(u_{qj} = l_{ti}(u_{tj}) + (q - t + \lfloor \Gamma/2 \rfloor) \times K$$

Implementation of ST-GCN

For a partitioning strategy with multiple subsets the adjacency matrix A consists of several matrices Aj where $A + I = \sum_j A_j$. Thus, ST-GCN in the single frame case can

be defined as:

$$f_{out} = \Lambda_j^{-1/2} A_j \Lambda_j^{-1/2} f_{in} W_j$$

where Λ is the diagonal degree matrix and W is the weight matrix.

However, the appearance of a joint in multiple body parts should have different significance in capturing the dynamics of these parts. Thus, we add a learnable mask M to every layer of spatial temporal graph convolution, which will scale the contribution of a nodes' feature to its neighboring nodes by a factor of the learned importance weight of each spatial graph edge in E_s. In the previous equation A_j is substituted with $A_j \otimes M$. Mask M is initialized as an all-one matrix.

ST-GCN Block architecture

The backbone network consists of a series of ST-GCN blocks. Both graph convolution and temporal convolution layers are concatenated to form a ST-GCN block that extracts both spatial and temporal information. The ST-GCN block also consists of other operations: batch normalization, which eliminates internal covariate shift and allows for much higher learning rates, ReLU and the residual block.

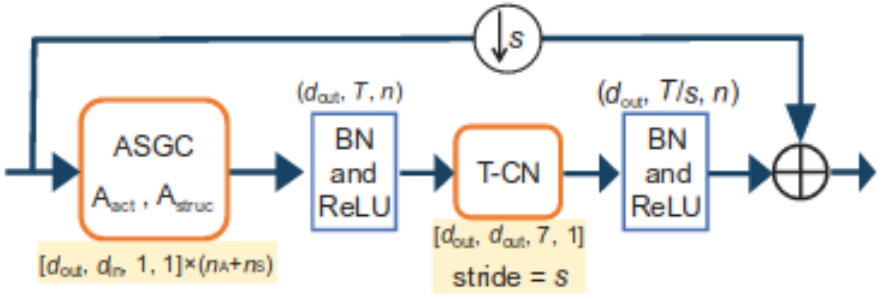


FIGURE 7.7: Structure of ST-GCN Block [50] .

7.4 Sign Language Recognition with 3D Information and Graph Convolutions

In our experiments we focus on how 3D pose and shape can be integrated effectively to our recognition pipeline so that they can enrich the model's knowledge about the articulated sign. Using the ST-GCN block as a main building module we build a network that combines the pose and shape parameters with visual features [30]. More specifically, the

Graph CNN architecture enables us to encode the skeletal structure within the network and leverage the spatial locality of the joints. Given an input frame a CNN encoder extracts image features from the input RGB representation. Consequently these features are are attached to the 2D and 3D hand and body joint coordinates.

The adapted ST-GCN network uses as input the 3D coordinates of each joint along with the input features and in each layer aggregates information from the spatiotemporal neighborhood of each node to obtain a per-vertex feature representation.

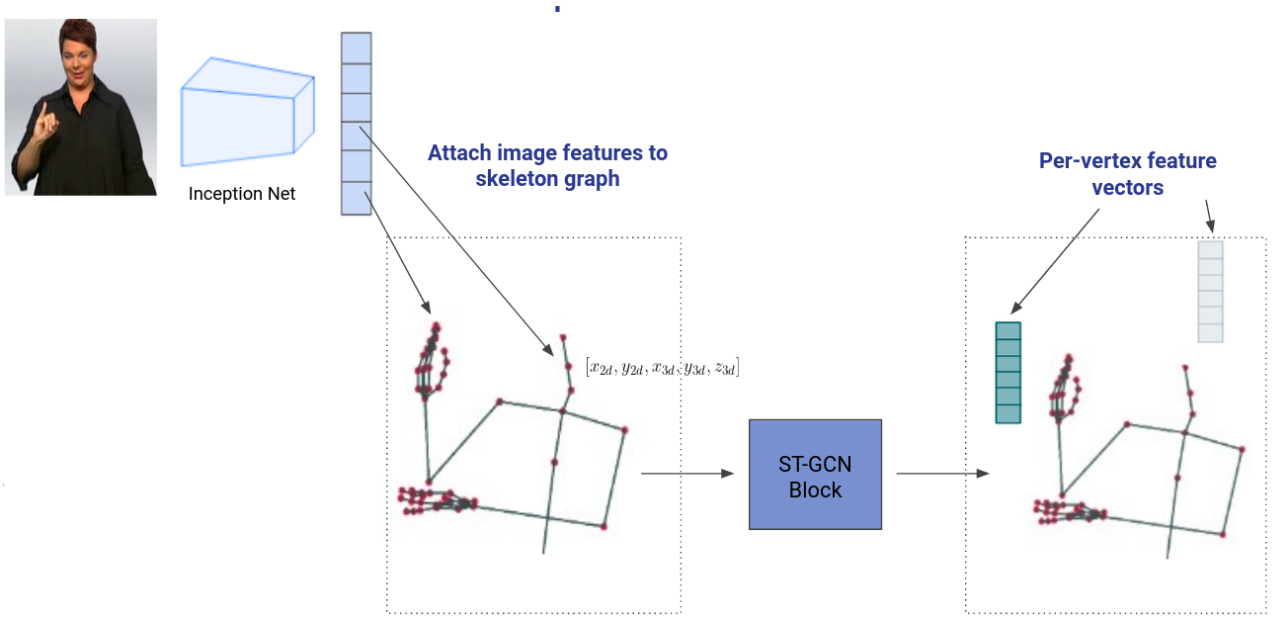


FIGURE 7.8: Overview of our proposed method

Model architecture

Our architecture consists of five stacked ST-GCN blocks with 256 channels each, followed by a global per node average pooling layer. The output is a 265-dim vector for each frame which is fed to a softmax layer for the final prediction. Inference is performed via a CTC beam search decoder with beam size = 3. For feature extraction we utilize an Inception Net, pretrained on sign language videos in a CNN-LSTM-HMM setup [5]. The complete architecture is outlined below.

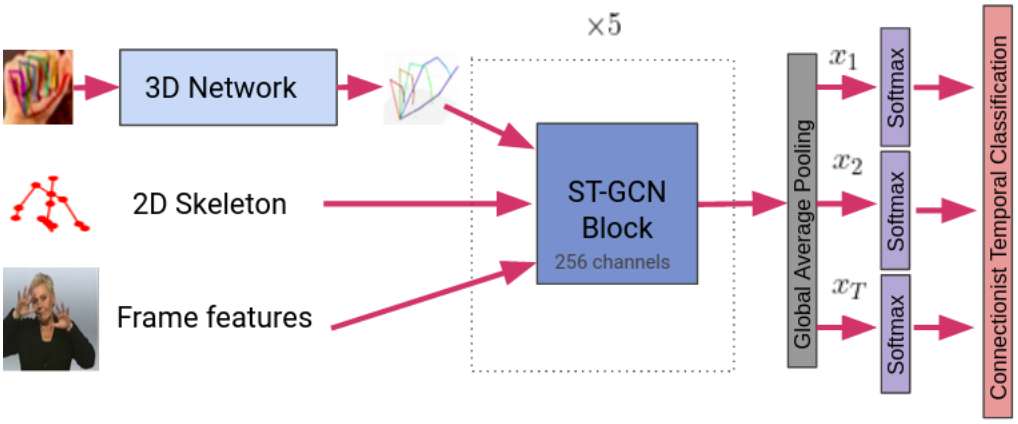


FIGURE 7.9: Model architecture

In action recognition tasks a successful approach is to train an ensemble of GCNs with inputs the positions of the joints and the bone vectors in two separate streams. This highlights not only the generalization abilities of the model but also how different pose representations can complement each other and yield increased performance. Thus, we follow a similar method and train a model with the same architecture in parallel but with the 6th dimensional pose parameterization θ extracted by ExPose embedded on each vertex/joint. We combine the softmax probabilities predicted by the two models and observe a significant reduction on Word Error Rate (WER).

7.4.1 Adaptive Graph Convolutions.

The graph convolution for the skeleton data described above is calculated based on a predefined graph, that follows the natural connectivity of the joints in the human body and hands. However, by implementing an adaptive graph convolutional layer we can overcome this limitation and optimize the topology of the graph during training [49]. To render the graph structure flexible, we change the spatial graph convolution to:

$$f_k = \sum_{k=1}^{K_u} W_k (A_k + B_k + C_k) f_{in}$$

where,

A_k is the normalized adjacency matrix.

B_k is an adjacency matrix, optimized during training. This allows the model to explore the existence and strength of connections between structurally distant joints.

C_k is a data-dependent graph which learns a unique topology for each sample.

More specifically, to determine the strength of the connection between two vertexes in C_k we rely on their similarity and apply the normalized embedded Gaussian function:

$$f(u_i, u_j) = \frac{e^{\theta(u_i)^T \theta(u_i)^T}}{\sum_{j=1}^{N} e^{\theta(u_i)^T \theta(u_i)^T}}$$

We first embed the input feature map into Ce×T×N with two embedding functions, i.e., θ and ϕ. The feature maps are reshaped to an N×CeT matrix and a CeT×N matrix and are multiplied to obtain an similarity matrix C_k, whose element C_{ijk} represents the similarity of vertex i and vertex j. Formally, C_k is calculated as follows:

$$C_k = softmax({f_{in}}^T {W_{\theta k}}^T W_{\phi k} f_{in})$$

,where W_θ and W_ϕ are the parameters of the embedding functions.

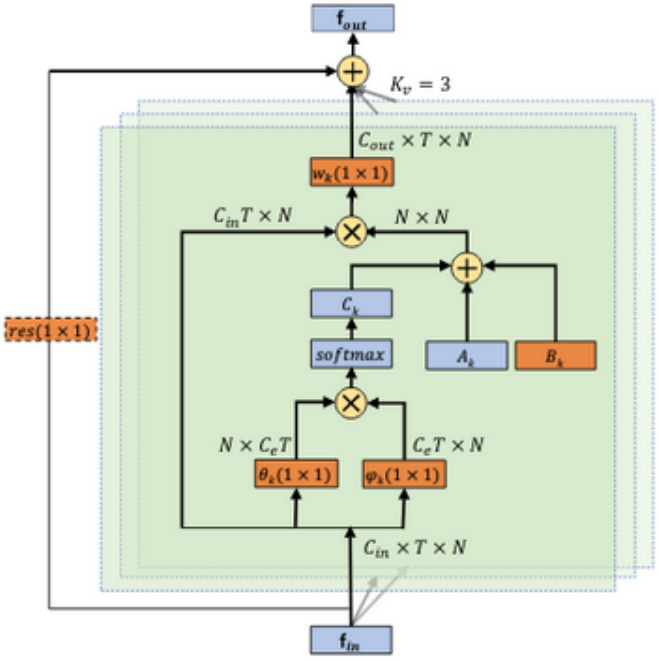


FIGURE 7.10: Adaptive graph convolutional layer [49]

7.4.2 Integration of a language model

The general objective of language models is to analyse bodies of text data and model the relationship between words in order to provide a basis for their word predictions. They

allow us to assign a conditional probability to every possible next word, generating a distribution over the entire vocabulary. In our problem a language model can provide context to distinguish between glosses/words with similar motion patterns. Intuitively, by taking into account language model predictions in the decoding phase we can improve the accuracy of our model.

A n-gram language model bases its predictions on a fixed preceding context of size N. Formally,

$$P(w_n|1_{1::n}) = P(w_n|w_{(n-N+1)})$$

RNN architectures relax this limitation via the hidden state, which embodies information about the preceding words back to the beginning of the sequence. In our experiments we use a 2-layer unidirectional GRU encoder as a language model with a 1024-dim hidden state and 1024-dim word embeddings. As an embedding layer we use a linear layer, which embeds each word into a continuous vector space. The input is a sequence of glosses and the output is this sequence shifted by one time step so that the model can learn to predict the next word in the sequence. As a loss function we use cross entropy:

$$L_{CE} = -\sum_{w \epsilon V} {y_w}^t \log \hat{y}_w^t$$

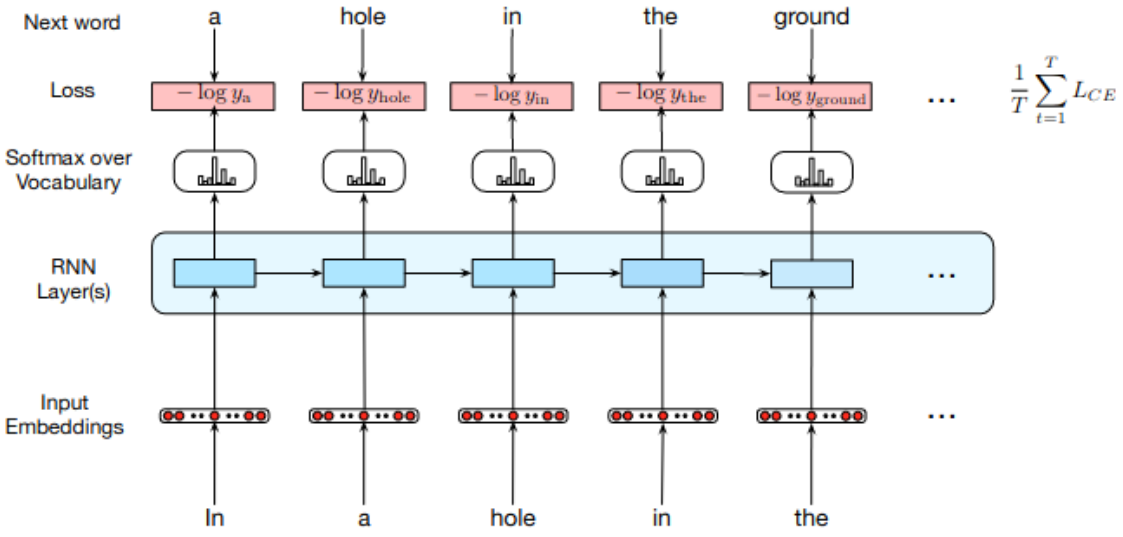


FIGURE 7.11: Recurrent Language Model Architecture [23].

The inference step in this case is performed by a label synchronous beam search decoding, that takes into consideration the language model predicted probabilities. The inference

objective is modified as :

$$\hat{C} = \arg\max_{h \epsilon \hat{\Omega}} \{\lambda \alpha_{ctc}(h, X) + (1 - \lambda)\alpha_{LM}(h, X)\}$$

where $\lambda = 0.9$

7.4.3 Training setup

We train our adapted ST-GCN model for 100 epochs and save the checkpoint which presents the lowest error rate on the dev set. We utilize plateau learning scheduling. Our initial learning rate is 0.001 and we decrease it by a factor of 0.7 if validation accuracy does not improve for 8 evaluation steps. Our optimizer is Adam and we use a batch size of 4 and a weight decay of 0.001. Training is performed on a RTX 2080 Ti GPU.

7.4.4 Adapted ST-GCN Experiments and comparison with baseline methods

We focus our experiments on the Phoenix 2014T dataset [4], which includes continuous sign language from 9 different signers with a vocabulary of 1066 different signs. The videos are adapted from daily news and weather forecast airings of the German public tv-station PHOENIX. Translations for these videos are provided in German spoken language. All recorded videos are at 25 frames per second and the size of the frames is 210 by 260 pixels.

FIGURE 7.12: Example frames from Phoenix 2014T dataset

TABLE 7.1: Comparison of performance of our adapted ST-GCN on Phoenix 2014T dataset [4] on dev and test set with various feature streams.

Methods	WER(Dev)	WER(Test)
Our ST-GCN(2D-Skeleton + adaptive)	45.7 %	46.5 %
Our ST-GCN(3D-Skeleton + adaptive)	50.1 %	50.3 %
Our ST-GCN(ExPose + adaptive)	44.0 %	44.8 %
Our ST-GCN(appearance feats + 2D-3D skeleton)	23.5 %	23.8 %
Our ST-GCN(appearance feats + ExPose)	24.1 %	24.1 %
Our ST-GCN(appearance feats + ExPose+ 2D-3D skeleton)	22.6 %	22.8 %
Our ST-GCN(all modalities + GRU language model)	- %	22.4 %

As a baseline we report the results attained by the methods described in the previous sections, namely sign language transformers and joint CTC/attention architectures.

TABLE 7.2: Comparison of performance of sign language transformers and joint CTC/attention architectures on [4] on test set with appearance feats as input.

Methods	WER(Test)
Sign Language Transformer(w/o label smoothing)	27.2 %
Sign Language Transformer(label smoothing)	25.7 %
Transformer-Hybrid CTC/Attention (location-aware attention)	25.5 %
Transformer-Hybrid CTC/Attention (location-aware/coverage attention)	25.2 %

Our method yields very promising results, surpassing many of the previous methods by a significant margin and performing on par with the current state-of-the-art approach of [60]. We observe that exploiting both joint position (skeleton) and rotation representation (ExPose) can significantly improve the accuracy of the model's predictions. By examining our results thoroughtly we draw the conclusion that our models' failed predictions often pertain to words with similar semantic meaning, such as RISIKO-GEFAHR (risk-danger), SUED-SUEDOST (south-southeast), JUNI-JULI (June-July). If we examine the embeddings of these words learned by our language model we confirm our intuition.

7.4.5 Sequential modeling with LSTMs

The temporal convolutional layers in the graph blocks succeed in extracting a representation of spatiotemporal neighborhoods and effectively capture short-term motions. However, modelling long-term motion dynamics is vital in sign language recognition. Thus, we benefit from the power of LSTMs, which can process long-range dependencies. In our case we implement a birectional LSTM encoder with two layers and a 256-dim hidden state. We choose a bidirectional property so as to exploit future context as well as previous context at the same time. The input to the encoder is the spatiotemporal latent representation extracted by the modified ST-GCN network and the output passed through a softmax linear layer for the final predictions. The modifications to our architecture are outlined below:

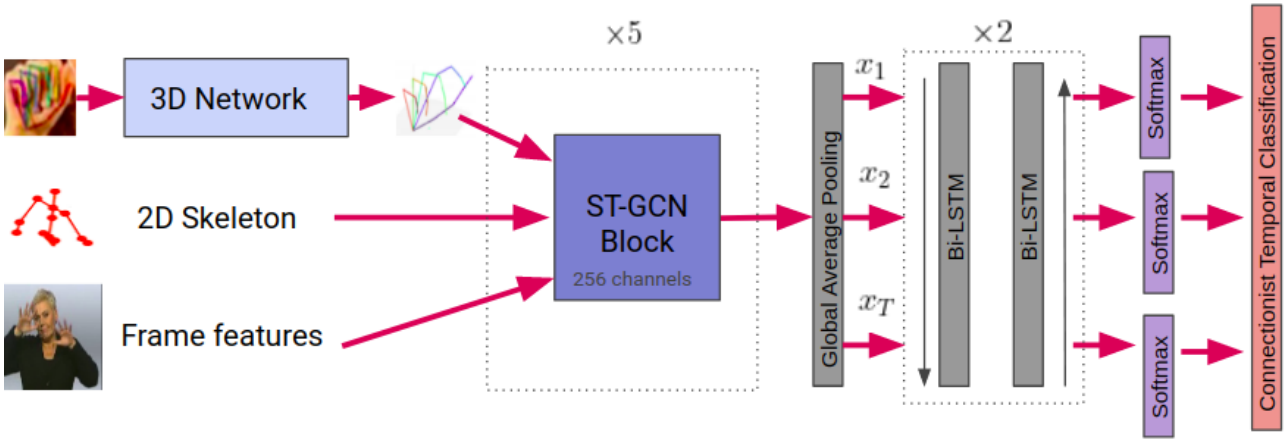


FIGURE 7.13: Complete network architecture with ST-GCN and Bi-LSTM layers.

7.4.5.1 Experiments with adapted ST-GCN-Bi-LSTM

TABLE 7.3: ST-GCN-Bi-LSTM performance on Phoenix 2014T dataset test set.

Methods	WER(Test)
ST-GCN-Bi-LSTM(appearance feats+2D-3D skeleton)	23.22 %
ST-GCN-Bi-LSTM(appearance feats+ExPose)	23.50 %

7.4.6 Posterior fusion with guiding methods

As it can be seen from the table above the modified network yields superior performance in both skeleton + appearance features and ExPose + appearance features setups. However, leveraging posterior fusion in this case is not a trivial task. CTC models emit very spiky posterior distributions where most frames emit the blank symbol and few frames emit the target symbol. As a result, CTC-LSTM models present non-aligned spike timings, which renders posterior fusion ineffective.

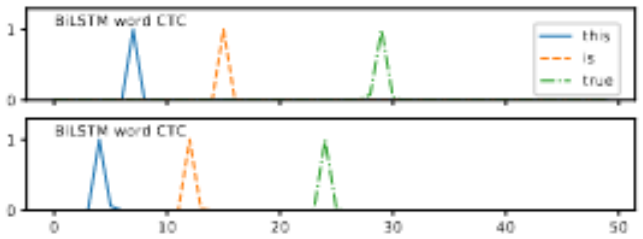


FIGURE 7.14: Illustrated example of non-aligned spike timings produced by word Bi-LSTM models in speech recognition. [32]

In order to address this problem, we implement a similar approach to [32] where a technique is proposed to explicitly guide the CTC spike timings to be aligned with those from a pre-trained CTC model (the guiding model). More specifically, when training a CTC model, in addition to the smoothed CTC loss, we incorporate a loss term that guides the spikes from the model being trained to occur at the same time as those from the guiding model. In this way models guided by the same guiding model present aligned spike timings.

When training the guided CTC model the posteriors for each time index predicted by the guiding model are converted to a mask M(X) by setting a 1 at the output symbol with the highest posterior and 0 at other symbols at each time index. When the blank symbol presents the highest posterior probability, we set 0 at this time index. This mask is applied to the symbols that the guiding CTC model outputs at each time step. More specifically, during the training process the posterior probabilities P(X) are predicted by the guided model. Through an elementwise multiplication of the mask and the posteriors, the masked posteriors $\hat{P}(X) = M(x) \odot P(X)$ are obtained. By maximizing this summation, the spike timings of the guided CTC model can be synchronised to those of the guiding CTC model.

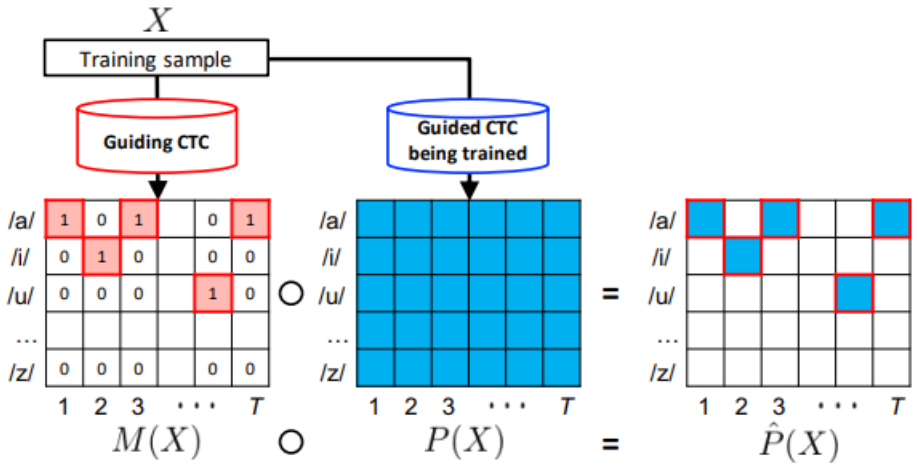


FIGURE 7.15: Diagram of proposed guided CTC training [32].

In our sign language experiments we use the model trained on 3D skeleton and appearance features as a guiding model and train a second guided model which uses ExPose and appearance feature as input with the method described above. The results of the posterior fusion are presented below:

We observe a significant improvement over the individual models, which was expected.

7.4.6.1 Experiments

TABLE 7.4: Performance of posterior fusion on Phoenix 2014T test set.

Methods	WER(Test)
2x ST-GCN-BLSTM	21.9 %

We also report our results in comparison with other published methods on this dataset.

TABLE 7.5: Performance comparison on Phoenix 2014T test set.

Methods	WER(Test)
STMC [60]	21.0 %
Our method	21.9 %
CrossModal[40]	24.3 %
Sign Language Transformer [5]	24.5 %
CNN-LSTM-HMM(f) [29]	26.0 %
CNN-LSTM-HMM (f+h+m) [29]	24.1 %
Stochastic SLR [38]	26.1 %

7.4.7 Connectionist Temporal Classification with Cross Entropy Regularization

Finally, we present an alternative to CTC loss, which we incorporated in our experiments but we did not observe noticeable gains. CTC tends to concentrate its output distribution on one specific path and sometimes fails to address ambiguous segmentation boundaries, e.g. action switching in continuous sign language recognition. [35] proposes a maximum conditional entropy based regularization for CTC, which prevents the entropy of feasible paths from decreasing rapidly and ensures that the probability is not dominated by a single path.

As we stated previously the conditional probability of a target sequence is defined as the sum of probabilities of all feasible paths. Namely,

$$p(l|X_{1:T}) = \sum_{\pi \epsilon B^{-1}(l)} \prod_{t=1}^{T} y_{\pi_t^t} \quad \forall \pi \epsilon L^T$$

CTC optimizes the loss function $L_{ctc} = -\log p(l|X_{1:T})$.

The error signal of CTC loss with respect to y_k^t is:

$$\frac{\partial L_{ctc}}{\partial y_k^t} = -\frac{1}{p(l|X) y_k^t} \sum_{\pi | \pi \epsilon B^{-1}(l), \pi_t = k} p(\pi|X)$$

where y_k^t denotes the probability vector of observing label k at time step t

We can infer that the error signal is proportional to the fraction of possible paths that end in symbol k at time t. Thus, once a path becomes dominant the error signal of $y_{\pi t}^t$ will dominate y_t at all time-steps causing CTC to focus on one path. In order to address this limitation the following regularization method is proposed, which prevents the entropy of the feasible paths from decreasing rapidly, increasing the model's generalization ability.

$$L_{enctc} = L_{ctc} - \beta H(p(\pi|l, X)$$

where H is the entropy of all feasible paths and it is defined as:

$$H(p(\pi|l, X)) = -\frac{1}{p(l|X)} \sum_{\pi \epsilon \beta^{-1}(l)} p(\pi|X) \log p(\pi|X) + \log p(l|X)$$

The error signal of entropy regularization term with respect to y_t is :

$$\frac{\partial Hp(\pi|L,x)}{\partial y_k^t} = \frac{Q(l)}{p(l|X)y_k^t}\left(\frac{\sum_{\{\pi|\pi\epsilon\beta^{-1}(l),\pi_t=k\}} p(\pi|X)\log p(\pi|X)}{Q(l)} - \frac{\sum_{\{\pi|\pi\epsilon\beta^{-1}(l),\pi_t=k\}} p(\pi|X)}{p(l|X)}\right)$$

where $Q(l) = \sum_{\{\pi|\pi\epsilon\beta^{-1}(l)\}} p(\pi|X)\log p(\pi|X)$

We observe that paths near the dominant path, mostly contribute the to the error signal. Thus, the error signal will increase the probability of nearby paths and encourage exploration during training.

Chapter 8

Summary and Future Directions

8.1 Summary

In this thesis we extensively studied the problem of sign language recognition and investigated the value of different architectures and multiple information streams.

In order to enrich the 3D human body representation we firstly examined 3D hand and body pose prediction methods. We proposed an approach that is based on deep linear architecture and enforces geometric constraints to "lift" 2D body and hand joint locations, extracted by the OpenPose framework to the 3D plane. This method yielded competitive results on both Rendered HandPose dataset and the FreiHand datset. We also provided qualitative examples that demonstrate that this approach can generalize effectively to "in the wild" sign language and gesture videos. We also examined semantic graph convolutions for the 3D "lifting" task and provided an overview of recent advances in 3D Hand Pose estimation. Additionally, in order to obtain full body shape and pose parameters we utilized the ExPose framework, which extracts body shape, pose and face expression parameters.

We exploited both 3D skeleton and the 3D mesh body parameterization, produced by the ExPose framework as well as frame appearance features in order to enhance the performance of our SLR models. Modeling the human body and hands as a graph we concatenated the 3D position, the 3D rotation vector and the frame appearance features as a representation for each node/joint. We combined the power of spatio-temporal graph convolutional networks and Bi-LSTM encoders for sequence modeling. To the best of our knowledge, an effective use of multiple-stream convolutional graph architecture for continuous sign language recognition has not been examined by other works.

To further enhance performance we proposed a guided posterior fusion of the models' predictions via a synchronization loss term. We achieved competitive performance on the widely researched Continuous Sign Language RWTH Phoenix 2014T dataset, surpassing the majority of state-of-the-art methods currently published.

We also compared our approach to strong baseline methods based on sign language transformers and lstm decoders with various attention mechanisms and reported our findings. We observed that graph convolutions can significantly improve recognition performance when pose and shape parameters are used as input. Additionally, we demonstrated that 3D position and rotation information boosts accuracy, highlighting the need for more expressive and detailed hand representations in sign language tasks.

Lastly, we briefly tackled the problem of isolated sign language recognition. More specifically, we experimented with skeleton-based deep learning methods, inspired by action and gesture recognition and reported our results on the GSLL dataset.

8.2 Future Directions

As far as Sign Language Recognition is concerned a possible approach would be to add richer information streams to our sign language recognition pipeline or extend our method to an online sign language recognition system.

- Depth information can be incorporated to our model by utilizing depth estimation systems even when the input is a monocular RGB frame. CNN-based object detection and saliency estimation tasks have demonstrated the added value of depth.
- 2D/3D optical flow features have been proven to provide action and sign language recognition models with meaningful insights and lead to higher accuracy.
- Integrate the method to an application for sign language recognition in real time.

In the task of 3D Pose Prediction, which we explored mainly in a SLR setting there are many promising ideas and directions.

- Novel architectures such as transformers can be successful in 2D to 3D pose "lifting" by exploiting the correlation between successive frames and modeling long-range dependencies of 2D pose sequences.

- Generative adversarial architectures can bridge the gap between synthetic and real hand pose data, yielding increased performance. Some novel works propose distillation of the 3D human pose structures from the fully annotated dataset to in-the-wild images with 2D pose annotations.

Bibliography

[1] Amorim, C. C. D., Macedo, D., & Zanchettin, C. (2019). Spatial-Temporal Graph Convolutional Networks for Sign Language Recognition. *Artificial Neural Networks and Machine Learning – ICANN 2019: Workshop and Special Sessions*.

[2] Antonakos, E., Pitsikalis, V., & Maragos, P. (2014). Classification of extreme facial events in sign language videos. *EURASIP Journal on Image and Video Processing, 2014:14*.

[3] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv:1803.01271*.

[4] Camgoz, N. C., Hadfield, S., Koller, O., Ney, H., & Bowden, R. (2018). Neural Sign Language Translation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[5] Camgoz, N. C., Koller, O., Hadfield, S., & Bowden, R. (2020). Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., & Sheikh, Y. A. (2019). OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[7] Cao, Z., Simon, T., Wei, S.-E., & Sheikh, Y. (2017). Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *CVPR*.

[8] Caridakis, G., Karpouzis, K., Drosopoulos, A., & Kollias, S. (2012). Non parametric, self organizing, scalable modeling of spatiotemporal inputs: The sign language paradigm. *Neural Networks*, *36*, 157–166. https://doi.org/https://doi.org/10.1016/j.neunet.2012.10.001

[9] Catalin Ionescu, C. S., Fuxin Li. (2011). Latent Structured Models for Human Pose Estimation. *International Conference on Computer Vision*.

[10] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[11] Chorowski, J., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-Based Models for Speech Recognition. *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, 577–585.

[12] Choutas, V., Pavlakos, G., Bolkart, T., Tzionas, D., & Black, M. J. (2020). Monocular Expressive Body Regression through Body-Driven Attention. *European Conference on Computer Vision (ECCV)*.

[13] Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, *1*, 886–893 vol. 1. https://doi.org/10.1109/CVPR.2005.177

[14] Graves, A., Ferńandez, S., Gomez, F., & Schmidhub, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. *ICML*.

[15] Hammond, D. K., Vandergheynst, P., & Gribonval, R. (2011). Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, *30*(2), 129–150. https://doi.org/https://doi.org/10.1016/j.acha.2010.04.005

[16] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition.

[17] He, Y., Hu, W., Yang, S.F., Qu, X., Wan, P., & Guo, Z. (2020). 3D hand pose estimation in the wild via graph refinement under adversarial learning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[18] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-term Memory. *Neural computation*, *9*, 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735

[19] Hou, J., Wang, G., Chen, X., Xue, J.-H., Zhu, R., & Yang, H. (2018). Spatial-Temporal Attention Res-TCN for Skeleton-Based Dynamic Hand Gesture Recognition. *ECCV Workshops*.

[20] Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, 448–456.

[21] Ionescu, C., Papava, D., Olaru, V., & Sminchisescu, C. (2014). Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[22] Joyce, J. M. (2011). Kullback-Leibler Divergence. In M. Lovric (Ed.), *International Encyclopedia of Statistical Science* (pp. 720–722). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-04898-2_327

[23] Jurafsky, D., & Martin, J. (2008). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (Vol. 2).

[24] Kim, S., Hori, T., & Watanabe, S. (2017). Joint CTC-attention based end-to-end speech recognition using multi-task learning. *ICASSP*, 4835–4839. https://doi.org/10.1109/ICASSP.2017.7953075

[25] Kim, T. S., & Reiter, A. (2017). Interpretable 3D Human Action Analysis with Temporal Convolutional Networks. *2017 Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

[26] Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. *International Conference on Learning Representations (ICLR)*.

[27] Klaeser, A., Marszalek, M., & Schmid, C. (2008). A Spatio-Temporal Descriptor Based on 3D-Gradients [doi:10.5244/C.22.99]. *Proceedings of the British Machine Vision Conference*, 99.1–99.10.

[28] Kläser, A., Marszalek, M., & Schmid, C. (2008). A Spatio-Temporal Descriptor Based on 3D-Gradients. *Proceedings of British Machine Vision Conference, 2008*. https://doi.org/10.5244/C.22.99

[29] Koller, O., Camgoz, N. C., Ney, H., & Bowden, R. (2020). 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*, *42*(9), 2306–2320. https://doi.org/10.1109/TPAMI.2019.2911077

[30] Kolotouros, N., Pavlakos, G., & Daniilidis, K. (2019). Convolutional Mesh Regression for Single-Image Human Shape Reconstruction. *CVPR*.

[31] Kratimenos, A., Pavlakos, G., & Maragos, P. (2021). Independent Sign Language Recognition with 3d Body, Hands, and Face Reconstruction. *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[32] Kurata, G., & Audhkhasi, K. (2019). Guiding CTC Posterior Spike Timings for Improved Posterior Fusion and Knowledge Distillation. *Proc. Interspeech 2019*, 1616–1620. https://doi.org/10.21437/Interspeech.2019-1952

[33] LeCun, Y., Haffner, P., Bottou, L., & Bengio, Y. (1999). Object Recognition with Gradient-Based Learning. *Shape, Contour and Grouping in Computer Vision* (pp. 319–345). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-46805-6_19

[34] Li, C., Zhong, Q., Xie, D., & Pu, S. (2018). Co-Occurrence Feature Learning from Skeleton Data for Action Recognition and Detection with Hierarchical Aggregation. *27th International Joint Conference on Artificial Intelligence*, 786–792.

[35] Liu, H., Jin, S., & Zhang, C. (2018). Connectionist Temporal Classification with Maximum Entropy Regularization. *Advances in Neural Information Processing Systems*, 837–847.

[36] Martinez, J., Hossain, R., Romero, J., & Little, J. J. (2017). A simple yet effective baseline for 3d human pose estimation. *ICCV*.

[37] Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., & Theobalt, C. (2018). GANerated Hands for Real-Time 3D Hand Tracking from Monocular RGB. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 49–59. https://doi.org/10.1109/CVPR.2018.00013

[38] Niu, Z., & Mak, B. (2020). Stochastic Fine-Grained Labeling of Multi-state Sign Glosses for Continuous Sign Language Recognition. *European Conference on Computer Vision*, 172–186.

[39] Ong, S., & Ranganath, S. (2005). Automatic Sign Language Analysis: A Survey and the Future beyond Lexical Meaning. *IEEE transactions on pattern analysis and machine intelligence*, *27*, 873–91. https://doi.org/10.1109/TPAMI.2005.112

[40] Papastratis, I., Dimitropoulos, K., Konstantinidis, D., & Daras, P. (2020). Continuous Sign Language Recognition Through Cross-Modal Alignment of Video and Text Embeddings in a Joint-Latent Space. *IEEE Access*, *8*, 91170–91180. https://doi.org/10.1109/ACCESS.2020.2993650

[41] Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A. A., Tzionas, D., & Black, M. J. (2019). Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[42] Pavlakos, G., Theodorakis, S., Pitsikalis, V., Katsamanis, A., & Maragos, P. (2014). Kinect-based multimodal gesture recognition using a two-pass fusion scheme. *2014 IEEE International Conference on Image Processing (ICIP)*, 1495–1499. https://doi.org/10.1109/ICIP.2014.7025299

[43] Pereyra, G., Tucker, G., Chorowski, J., Kaiser, L., & Hinton, G. E. (2017). Regularizing Neural Networks by Penalizing Confident Output Distributions. *ArXiv*, *abs/1701.06548*.

[44] Pitsikalis, V., Theodorakis, S., Vogler, C., & Maragos, P. (2011). Advances in Phonetics-based Sub-Unit Modeling for Transcription Alignment and Sign Language Recognition. *IEEE Conf. on Computer Vision & Pattern Recognition Workshops*.

[45] Pu, J., Zhou, W., & Li, H. (2018). Dilated Convolutional Network with Iterative Optimization for Continuous Sign Language Recognition. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*.

[46] Roussos, A., Theodorakis, S., Pitsikalis, V., & Maragos, P. (2013). Dynamic Affine-Invariant Shape-Appearance Handshape Features and Classification in Sign Language Videos. *Journal of Machine Learning Research*, *14*, 1627–1663.

[47] Rumelhart, D. E., & McClelland, J. L. (1987). Learning Internal Representations by Error Propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations* (pp. 318–362).

[48] See, A., Liu, P., & Manning, C. (2017). Get To The Point: Summarization with Pointer-Generator Networks. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 1073–1083. https://doi.org/10.18653/v1/P17-1099

[49] Shi, L., Zhang, Y., Cheng, J., & Lu, H. (2019). Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. *CVPR*.

[50] Sijie Yan, D. L., Yuanjun Xiong. (2018). Spatial-Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. *AAAI*.

[51] Simon, T., Joo, H., Matthews, I., & Sheikh, Y. (2017). Hand Keypoint Detection in Single Images using Multiview Bootstrapping. *CVPR*.

[52] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.*, *15*(1), 1929–1958.

[53] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2014). Going Deeper with Convolutions.

[54] Theodorakis, S., Pitsikalis, V., & Maragos, P. (2014). Dynamic–static unsupervised sequentiality, statistical subunits and lexicon for sign language recognition. *Image and Vision Computing*, *32*(8), 533–549. https://doi.org/https://doi.org/10.1016/j.imavis.2014.04.012

[55] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *Conference on Neural Information Processing Systems (NIPS 2017)*.

[56] Vogler, C., & Metaxas, D. (2001). A Framework for Recognizing the Simultaneous Aspects of American Sign Language. *Computer Vision and Image Understanding*, *81*(3), 358–384. https://doi.org/https://doi.org/10.1006/cviu.2000.0895

[57] Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local Neural Networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7794–7803. https://doi.org/10.1109/CVPR.2018.00813

[58] Watanabe, S., Hori, T., Kim, S., Hershey, J. R., & Hayashi, T. (2017). Hybrid CTC/Attention Architecture for End-to-End Speech Recognition. *IEEE Journal of Selected Topics in Signal Processing*, *11*(8), 1240–1253. https://doi.org/10.1109/JSTSP.2017.2763455

[59] Zhao, L., Peng, X., Tian, Y., Kapadia, M., & Metaxas, D. N. (2019). Semantic Graph Convolutional Networks for 3D Human Pose Regression. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3420–3430. https://doi.org/10.1109/CVPR.2019.00354

[60] Zhou, H., Zhou, W., Zhou, Y., & Li, H. (2020). Spatial-Temporal Multi-Cue Network for Continuous Sign Language Recognition. *AAAI*.

[61] Zimmermann, C., & Brox, T. (2017). Learning to Estimate 3D Hand Pose from Single RGB Images [https://arxiv.org/abs/1705.01389]. *International Conference on Computer Vision (ICCV)*.

[62] Zimmermann, C., Ceylan, D., Yang, J., Russel, B., Argus, M., & Brox, T. (2019). FreiHAND: A Dataset for Markerless Capture of Hand Pose and Shape from Single RGB Images. *International Conference on Computer Vision (ICCV)*.