



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

***ΑΝΑΓΝΩΡΙΣΗ ΕΠΙΚΙΝΔΥΝΩΝ
ΚΑΤΑΣΤΑΣΕΩΝ ΚΑΤΑ ΤΗΝ ΟΔΗΓΗΣΗ***

ΚΑΡΑΓΙΑΝΝΗΣ ΔΗΜΗΤΡΙΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΚΟΛΟΜΒΑΤΣΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ
ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΗΣ

Λαμία, έτος 2026

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ.), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 22/08/2026

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΠΕΡΙΛΗΨΗ

Στην σημερινή εποχή ο ιδιαίτερα αυξανόμενος αριθμός οχημάτων στους δρόμους και κατά συνέπεια των οδηγών καθιστά την διασφάλιση των ασφαλών συνθηκών οδήγησης καίριας σημασίας πιθανόν όσο ποτέ άλλοτε. Κινούμενοι προς αυτήν την κατεύθυνση οι παράμετροι και οι συνθήκες που πρέπει να λαμβάνονται υπόψιν τόσο εντός του οχήματος όσο και εκτός στο οδικό δίκτυο είναι πολλές με την κάθε μια να έχει την δική της σημασία και βαρύτητα. Ένας παράγοντας με μεγάλη σημασία, βαρύτητα και αντίκτυπο στην ασφαλή οδήγηση είναι αυτός της κόπωσης, υπνηλίας του οδηγού του οχήματος που μπορεί να τον οδηγήσει σε ύπνο και κατά συνέπεια δυσάρεστες συνέπειες, καθώς χάνει την επαφή με το περιβάλλον του, τόσο για τον ίδιο όσο και για τους υπόλοιπους οδηγούς στο οδικό δίκτυο όπως πιθανόν και για τους πεζούς. Η συγκεκριμένη εργασία είναι μια προσπάθεια επίλυσης αυτού του προβλήματος προτείνοντας ένα σύστημα το οποίο κάνοντας χρήση προχωρημένων τεχνικών μηχανικής μάθησης, τεχνικών επεξεργασίας δεδομένων και τους μαθηματικούς λόγους EAR και MAR, οι οποίοι είναι δεκαδικοί αριθμοί που εκφράζουν πόσο ανοιχτά είναι τα μάτια και το στόμα αντίστοιχα και θα περιγραφούν αναλυτικότερα αργότερα, προβλέπει καταστάσεις υπνηλίας στον οδηγό. Το τεχνητό νευρωνικό δίκτυο που δημιουργήθηκε για αυτόν τον σκοπό αποτελείται από λίγες χιλιάδες παραμέτρους, δίνοντάς του την δυνατότητα να εγκατασταθεί και να χρησιμοποιηθεί ακόμα και από συσκευές με περιορισμένους πόρους προσφέροντας του ευελιξία και προσαρμοστικότητα.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Υπνηλία, Ασφάλεια, Δεδομένα, Επεξεργασία, EAR, MAR, Προχωρημένες τεχνικές μηχανικής μάθησης

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ	I
<u>1.ΕΙΣΑΓΩΓΗ ΚΑΙ ΚΙΝΗΤΡΟ.....</u>	<u>3</u>
1.1: ΤΙ ΕΙΝΑΙ Η ΥΠΗΛΙΑ, ΓΕΝΙΚΕΣ ΠΛΗΡΟΦΟΡΙΕΣ ΚΑΙ ΣΤΑΤΙΣΤΙΚΑ ΣΤΟΙΧΕΙΑ	3
1.2: ΤΙ ΕΙΝΑΙ Η ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ ΚΑΙ ΙΣΤΟΡΙΚΑ ΣΤΟΙΧΕΙΑ	4
1.3: ΤΙ ΕΙΝΑΙ Η ΒΑΘΙΑ ΜΑΘΗΣΗ, ΓΕΝΙΚΕΣ ΠΛΗΡΟΦΟΡΙΕΣ.....	5
1.4: ΔΟΜΗ ΤΗΣ ΠΑΡΟΥΣΑΣ ΕΡΓΑΣΙΑΣ, ΛΙΓΑ ΛΟΓΙΑ	7
<u>2. ΠΡΟΧΩΡΗΜΕΝΕΣ ΤΕΧΝΙΚΕΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ.....</u>	<u>9</u>
2.1: ΠΙΟ ΑΝΑΛΥΤΙΚΗ ΕΙΣΑΓΩΓΗ ΣΤΑ ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	9
2.2: ΛΕΙΤΟΥΡΓΙΑ ΤΩΝ ΤΕΧΝΗΤΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ.....	10
2.2.1: ΕΜΠΡΟΣ ΔΙΑΔΟΣΗ (FORWARD PROPAGATION)	10
2.2.2: ΠΙΣΩ ΔΙΑΔΟΣΗ (BACKPROPAGATION)	14
2.2.3: ΕΠΑΝΑΛΗΠΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ	18
2.3: ΣΥΝΕΛΙΚΤΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ (CNN).....	18
2.4: LONG SHORT-TERM MEMORY (LSTM)	20
2.5: TENSORFLOW ΚΑΙ KERAS, ΛΙΓΑ ΛΟΓΙΑ.....	22
<u>3.ΕΦΑΡΜΟΓΕΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΓΙΑ ΑΝΑΓΝΩΡΙΣΗ ΥΠΗΛΙΑΣ..</u>	<u>24</u>
3.1: ΤΑΞΙΝΟΜΗΣΗ ΕΦΑΡΜΟΓΩΝ ΑΝΑΓΝΩΡΙΣΗΣ ΥΠΗΛΙΑΣ.....	24
3.2: ΚΛΑΣΣΙΚΕΣ ΜΕΘΟΔΟΙ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ, ΟΡΟΣΗΜΑ & ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ..	25
3.3: ΑΝΑΦΟΡΑ ΜΕΘΟΔΩΝ ΒΑΘΙΑΣ ΜΑΘΗΣΗΣ.....	26
3.4: ΣΥΓΚΡΙΣΗ CNNs ΕΝΑΝΤΙ LSTMs ΣΤΗΝ ΑΝΑΓΝΩΡΙΣΗ ΤΗΣ ΥΠΗΛΙΑΣ	27
3.5: DATASETS, ΜΕΤΡΙΚΕΣ ΑΞΙΟΛΟΓΗΣΗΣ ΚΑΙ ΥΠΑΡΚΤΕΣ ΠΡΟΚΛΗΣΕΙΣ.....	28
3.5.1: ΑΝΑΦΟΡΑ ΣΤΑ ΠΙΟ ΓΝΩΣΤΑ ΔΗΜΟΣΙΑ ΣΥΝΟΛΑ ΔΕΔΟΜΕΝΩΝ	28
3.5.2: ΣΗΜΑΝΤΙΚΟΤΗΤΑ DATASET & ΜΕΤΡΙΚΕΣ ΑΞΙΟΛΟΓΗΣΗΣ	29
3.5.3: ΥΠΑΡΚΤΕΣ ΠΡΟΚΛΗΣΕΙΣ ΣΤΗΝ ΑΝΑΓΝΩΡΙΣΗ ΤΗΣ ΥΠΗΛΙΑΣ	30
3.6: ΤΟΠΟΘΕΤΗΣΗ ΤΗΣ ΠΑΡΟΥΣΑΣ ΕΡΓΑΣΙΑΣ	31
<u>4. ΤΟ ΠΡΟΤΕΙΝΟΜΕΝΟ ΣΥΣΤΗΜΑ.....</u>	<u>32</u>
4.1: ΦΑΣΗ 1 - ΔΗΜΙΟΥΡΓΙΑ ΑΡΧΕΙΟΥ ΚΕΙΜΕΝΟΥ ΑΠΟ ΤΑ ΒΙΝΤΕΟ ΤΟΥ DATASET	32
4.1.1: ΠΕΡΙΓΡΑΦΗ ΜΑΘΗΜΑΤΙΚΩΝ ΛΟΓΩΝ EAR & MAR.....	32
4.1.2: ΔΙΑΔΙΚΑΣΙΑ ΔΗΜΙΟΥΡΓΙΑΣ ΑΡΧΕΙΟΥ ΚΕΙΜΕΝΟΥ	33
4.2: ΦΑΣΗ 2 – ΠΡΟΕΤΟΙΜΑΣΙΑ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΤΟ TNN	37
4.2.1: ΕΙΣΑΓΩΓΗ	37
4.2.2: ΔΙΑΧΩΡΙΣΜΟΣ ΔΕΔΟΜΕΝΩΝ ΣΕ ΥΠΟΣΥΝΟΛΑ.....	38
4.2.3: ΔΗΜΙΟΥΡΓΙΑ ΧΡΟΝΙΚΩΝ ΠΑΡΑΘΥΡΩΝ & ΕΤΙΚΕΤΩΝ ΓΙΑ ΤΟ TNN	39
4.3: ΦΑΣΗ 3 – ΑΡΧΙΤΕΚΤΟΝΙΚΗ TNN ΚΑΙ ΕΚΠΑΙΔΕΥΣΗ.....	41
4.3.1: ΠΑΡΟΥΣΙΑΣΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΠΡΟΤΕΙΝΟΜΕΝΟΥ ΜΟΝΤΕΛΟΥ ..	41
4.3.2: ΠΑΡΟΥΣΙΑΣΗ ΛΕΠΤΟΜΕΡΕΙΩΝ ΕΚΠΑΙΔΕΥΣΗΣ	44

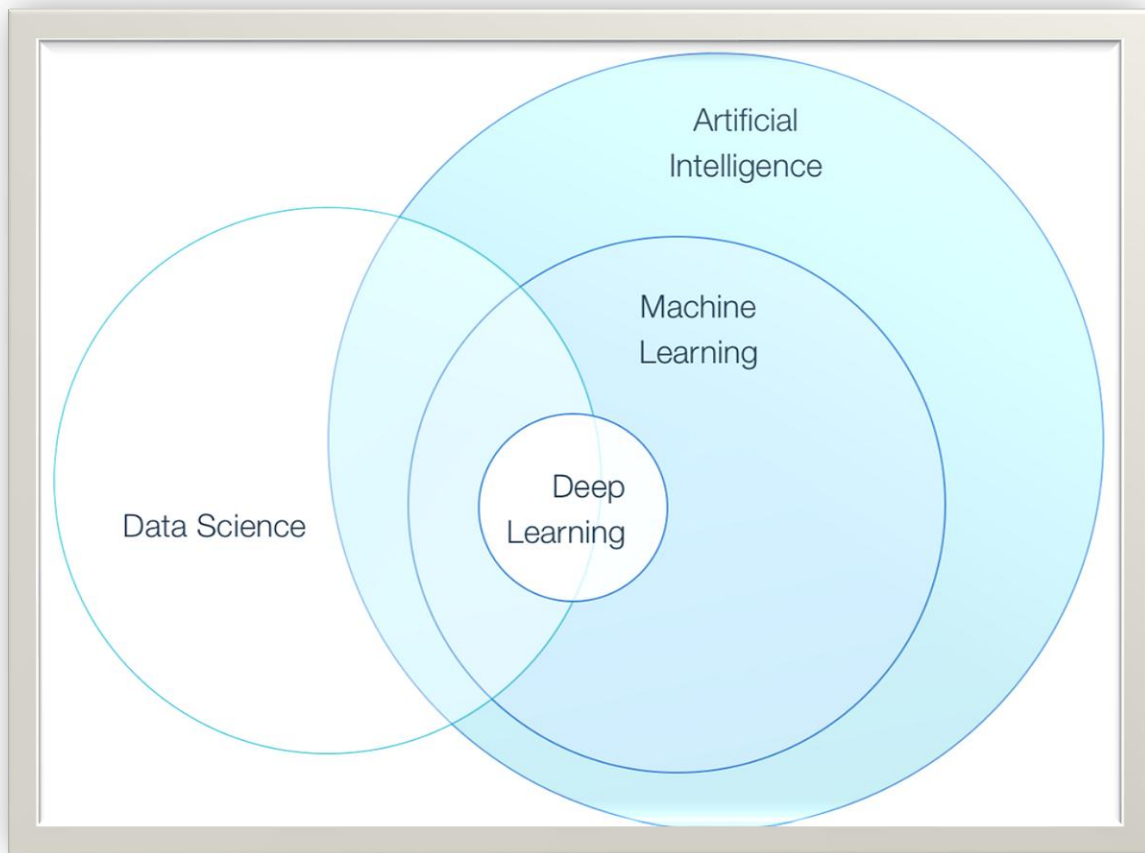
5. ΠΕΙΡΑΜΑΤΙΚΗ ΑΠΟΤΙΜΗΣΗ	47
5.1: ΤΟ ΣΥΝΟΛΟ ΔΕΔΟΜΕΝΩΝ ΤΗΣ ΕΡΓΑΣΙΑΣ	47
5.2: ΠΛΗΡΟΦΟΡΙΕΣ ΔΙΑΧΩΡΙΣΜΟΥ DATASET & ΔΗΜΙΟΥΡΓΙΑΣ ΠΑΡΑΘΥΡΩΝ.....	48
5.3: ΜΕΤΡΙΚΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ.....	49
5.4: ΕΥΡΕΣΗ ΟΡΙΟΥ ΠΙΘΑΝΟΤΗΤΩΝ & ΕΞΟΜΑΛΥΝΣΗ ΑΥΤΩΝ	51
5.5: ΠΕΙΡΑΜΑΤΙΣΜΟΣ & ΤΕΛΙΚΟ ΠΡΟΤΕΙΝΟΜΕΝΟ ΣΥΣΤΗΜΑ.....	52
5.6: ΠΑΡΟΥΣΙΑΣΗ LIVE DEMO ΕΡΓΑΣΙΑΣ	57
6. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΕΚΤΑΣΕΙΣ.....	62
6.1: ΑΝΑΦΟΡΑ ΣΥΜΠΕΡΑΣΜΑΤΩΝ ΚΑΙ ΠΡΟΕΚΤΑΣΕΩΝ	62
6.2: ΑΠΟΘΕΤΗΡΙΟ ΚΩΔΙΚΑ ΣΤΟ GITHUB.....	63
7.ΒΙΒΛΙΟΓΡΑΦΙΑ	64

1.Εισαγωγή και Κίνητρο

1.1: Τι είναι η υπνηλία, γενικές πληροφορίες και στατιστικά στοιχεία

Η υπνηλία είναι μια κατάσταση κατά την οποία το άτομο διακατέχεται από έντονη επιθυμία για να κοιμηθεί και μπορεί να οφείλεται σε πολλές διαφορετικές αιτίες. Ακόμα πολλές φορές στο άτομο που εκείνη την στιγμή πάσχει από υπνηλία μπορεί να παρουσιαστεί ληθαργία, γενική αδυναμία και απουσία ή μείωση της πνευματικής ευκινησίας, οξύτητας, συγκέντρωσης και εγρήγορσης. Άρα αυτομάτως θα μπορούσε κάποιος να διαπιστώσει-συμπεράνει γιατί αποτελεί σοβαρό κίνδυνο όταν εμφανίζεται σε έναν οδηγό κατά την οδήγηση και πόσο σημαντικό και κρίσιμο είναι να υπάρχουν τα κατάλληλα μέτρα και συστήματα για την αντιμετώπιση της, μέσω της πρόβλεψης, προς όφελος της γενικής οδηγικής ασφάλειας. Περνώντας σε περισσότερες λεπτομέρειες σχετικά με το φαινόμενο της υπνηλίας μερικές αιτίες που μπορούν να το προκαλέσουν είναι η διαταραχή του κερκάδιου ρυθμού, το άγχος, η ανησυχία, η κούραση, διαταραχές της ψυχολογίας του ατόμου όπως είναι η κατάθλιψη κ.α. [1]. Αν μπορούσαμε να συνοψίσουμε τους λόγους που προκαλούν αυτό το φαινόμενο θα λέγαμε ότι η υπνηλία οφείλεται είτε στον προβληματικό τρόπο ζωής του ατόμου είτε σε κάποια πάθηση. Σε αυτό εδώ το σημείο αξίζει να αναφερθούν κάποια στατιστικά στοιχεία τα οποία συγκέντρωσε ο μεγαλύτερος φιλανθρωπικός οργανισμός υγείας της Μεγάλης Βρετανίας έπειτα από έρευνα που πραγματοποίησε με την συμμετοχή 8.000 ενήλικων οδηγών και τα οποία παρουσιάζονται στο [2]. Σύμφωνα λοιπόν με αυτήν την έρευνα σχεδόν το 50% των οδηγών στην ηλικιακή ομάδα από 35 έως 44 έτη κοιμάται κάθε βράδυ μόλις 5-6 ώρες ενώ από την άλλη πλευρά τις συνιστώμενες καθημερινές ώρες ύπνου δηλαδή 7-8 συγκεντρώνει μόνο το 33% των συμμετεχόντων της έρευνας. Κατά συνέπεια αυτά τα δεδομένα μπορούν να επιφέρουν κόπωση και υπνηλία στους οδηγούς, ιδιαίτερα κατά την διάρκεια της νύχτας και σε μακρινά ταξίδια, όταν βρίσκονται πίσω από το τιμόνι γεγονός που αυξάνει δραματικά τον κίνδυνο πρόκλησης αλλά και τα τροχαία ατυχήματα που τελικά συμβαίνουν εξαιτίας της υπνηλίας, τα οποία πολλές φορές στοιχίζουν ανθρώπινες ζωές. Χαρακτηριστικό είναι επίσης το στατιστικό από την ίδια έρευνα σύμφωνα με το οποίο μόνο στην Αγγλία 300 άνθρωποι έχασαν την ζωή τους, επειδή αποκοιμήθηκαν κατά την οδήγηση, σε μόλις έναν χρόνο. Τέλος ενδιαφέρον εμφανίζουν μερικά ακόμα στατιστικά από έρευνα που πραγματοποίησε η λέσχη αυτοκινήτου AA (Automobile Association) συλλέγοντας δεδομένα από ένα σύνολο 20.000 ενήλικων οδηγών, τα οποία αναφέρονται επίσης στο [2], και σύμφωνα με αυτά το 13% του συνόλου των ερωτηθέντων με ειλικρίνεια παραδέχτηκε στην ερώτηση αν έχει αποκοιμηθεί ποτέ κατά την διάρκεια της οδήγησης ότι το έχει κάνει. Επίσης το σημαντικό ποσοστό του 40% του συνόλου δήλωσε πως είχε τύχει να ένιωσε τόσο μεγάλη κόπωση και υπνηλία κατά την διάρκεια της οδήγησης που θα του ήταν αδύνατον να καταφέρει να μείνει ξύπνιο και σε εγρήγορση καθόλη την διάρκεια της διαδρομής που εκτελούσε. Όλα αυτά τα στατιστικά στοιχεία υποδηλώνουν μια γενική χειροτέρευση της ποιότητας του ύπνου των σημερινών ανθρώπων και της αύξησης της κόπωσης, καθώς και τις επιπτώσεις αυτού, ενισχύοντας την θέση που εκφράζει την

1.3: Τι είναι η βαθιά μάθηση, γενικές πληροφορίες



Εικόνα 1: Κατηγοριοποίηση Τεχνητής Νοημοσύνης

Το επιστημονικό πεδίο της βαθιάς μάθησης, πάνω στο οποίο βασίστηκε και η παρούσα εργασία, αποτελεί τμήμα του ευρύτερου πεδίου της τεχνητής νοημοσύνης όπως φαίνεται και από την παραπάνω εικόνα. Η βαθιά μάθηση αποτελεί το πιο σύγχρονο και τεχνολογικά εξελιγμένο πεδίο της τεχνητής νοημοσύνης με την βοήθεια του οποίου οι υπολογιστές μαθαίνουν, καταλαβαίνουν και αλληλοεπιδρούν καλύτερα και πιο αποτελεσματικά με σύνθετα δεδομένα. Το συγκεκριμένο επιστημονικό πεδίο εισάγει, πέραν όλων των άλλων, 2 πολύ βασικές, δομικές θα λέγαμε για το πεδίο έννοιες, αυτή των τεχνητών νευρωνικών δικτύων, τα οποία μιμούνται την δομή και την λειτουργία των νευρωνικών δικτύων του εγκεφάλου με απώτερο σκοπό να πλησιάσουν στην ανθρώπινη νοημοσύνη και αυτή των τεχνητών νευρώνων, οι οποίοι παρομοίως μιμούνται την λειτουργία των βιολογικών νευρώνων, αποτελούν τα δομικά στοιχεία των τεχνητών νευρωνικών δικτύων, καταρτίζονται σε στρώματα σε αυτό και “εκπαιδεύονται” με τελικό στόχο την επεξεργασία δεδομένων. Το επίθετο βαθιά που χαρακτηρίζει αυτού του είδους την μάθηση για υπολογιστικά συστήματα αναφέρεται στην χρήση πολλών στρωμάτων τεχνητών νευρώνων σε τέτοιου είδους δίκτυα (έως αρκετές εκατοντάδες ή χιλιάδες στρώματα). Τώρα σε ότι αφορά τις μεθόδους με τις οποίες αυτά τα δίκτυα και κατά συνέπεια οι νευρώνες εκπαιδεύονται αυτές μπορούν να είναι είτε εποπτευόμενες ή ημι-εποπτευόμενες ή μη εποπτευόμενες. Η πιο απλή

αρχιτεκτονική ενός τέτοιου τεχνητού δικτύου είναι τα τεχνητά νευρωνικά δίκτυα προώθησης (Feedforward Neural Networks) όπου τα δεδομένα μπορούν να “κινούνται” μέσα στο δίκτυο μόνο προς μία κατεύθυνση, από την είσοδο στην έξοδο αυτού, χρησιμοποιούνται για βασικές διεργασίες όπως είναι η κατηγοριοποίηση δεδομένων. Μια άλλη ευρέως γνωστή και πολύ χρησιμοποιούμενη αρχιτεκτονική πλέον είναι αυτή των συνελικτικών νευρωνικών δικτύων (Convolutional Neural Networks) τα οποία χρησιμοποιούνται κυρίως για την επεξεργασία δεδομένων σε μορφή πινάκων, όπως είναι οι εικόνες. Τα αναδρομικά νευρωνικά δίκτυα (Recurrent Neural Networks) αποτελούν επίσης μια πολύ χρήσιμη αρχιτεκτονική με την βοήθεια της οποίας επεξεργαζόμαστε ακολουθίες δεδομένων όπως είναι οι χρονοσειρές και η φυσική γλώσσα των ανθρώπων. Σημαντική είναι επίσης η αρχιτεκτονική των παραγωγικών ανταγωνιστικών δικτύων (Generative Adversarial Networks) τα οποία είναι ιδιαίτερα αποτελεσματικά στην δημιουργία εικόνων, στην επαύξηση δεδομένων κ.α. Αυτές είναι μερικές από τις αρχιτεκτονικές που έχουν προταθεί ανά τα χρόνια, ενισχύοντας τις δυνατότητες αλλά και τις προοπτικές της βαθιάς μάθησης και κατά συνέπεια της τεχνητής νοημοσύνης. Η βαθιά μάθηση εξαιτίας των αυξημένων δυνατοτήτων της βρίσκει εφαρμογές σε κρίσιμους και διάφορους τομείς της επιστήμης των υπολογιστών. Ανάμεσα τους είναι η υπολογιστική όραση, όπου τεχνητά νευρωνικά δίκτυα δίνουν την δυνατότητα στους υπολογιστές να αναγνωρίζουν και να καταλαβαίνουν δεδομένα εικόνες, η επεξεργασία φυσικής γλώσσας όπου με την χρήση τέτοιων δικτύων οι υπολογιστές μπορούν να καταλαβαίνουν και να παράγουν την ανθρώπινη γλώσσα τόσο σε επίπεδο κειμένου αλλά και ομιλίας, η ενισχυτική μάθηση που εισάγει την έννοια των ευφύων πρακτόρων κ.α. Συμπερασματικά τα τεχνητά νευρωνικά δίκτυα χαρακτηρίζονται από πολλά πλεονεκτήματα έναντι της κλασσικής μηχανικής μάθησης, όπως είναι για παράδειγμα η αυξημένη ακρίβεια τους σε πολλές διεργασίες ή η ικανότητα τους να μαθαίνουν από μόνα τους τα σημαντικά χαρακτηριστικά και μοτίβα των δεδομένων με τα οποία τροφοδοτούνται χωρίς να υπάρχει η ανάγκη της ανθρώπινης παρέμβασης, όπως συμβαίνει με την κλασσική μηχανική μάθηση, όπου χρειάζεται ένας άνθρωπος να διαλέξει ποια χαρακτηριστικά των δεδομένων έχουν μεγαλύτερη σημασία ώστε μαζί με τα ίδια τα δεδομένα να τροφοδοτήσουν τον αλγόριθμο μηχανικής μάθησης, κ.α. Βέβαια το ότι υπερέχουν σε πολλές διεργασίες έναντι της κλασσικής μηχανικής μάθησης δεν σημαίνει ότι δεν έχουν μειονεκτήματα και χαρακτηριστικά που επιδέχονται βελτίωση, για παράδειγμα ένα σημαντικό μειονέκτημα τους είναι ότι σε αντίθεση με τους αλγορίθμους της κλασσικής μηχανικής μάθησης απαιτούν μεγάλο όγκο δεδομένων για να “εκπαιδευτούν” σωστά και να μπορέσουν αυτόματα να μάθουν σημαντικά μοτίβα από αυτά, ο οποίος όγκος πολλές φορές είναι δύσκολο να συγκεντρωθεί π.χ. ιατρικά δεδομένα, επίσης η “εκπαίδευση” τους πολλές φορές είναι πολύ απαιτητική υπολογιστικά, ακόμα η αρχιτεκτονική και η λειτουργία τους είναι πολύ σύνθετη και δύσκολο να κατανοηθεί ενώ τα αποτελέσματα που δίνουν πολλές φορές είναι ιδιαίτερα απαιτητικό να ερμηνευθούν σε αντίθεση με τους αλγορίθμους της κλασσικής μηχανικής μάθησης που είναι πιο εύκολο τόσο να κατανοηθούν αλλά και να εξηγηθούν, κ.α.

1.4: Δομή της παρούσας εργασίας, λίγα λόγια

Μέχρι στιγμής έχει δοθεί μια σύντομη εισαγωγική περίληψη για την παρούσα πτυχιακή και έχει παρουσιαστεί το πρώτο κεφάλαιο αυτής με τίτλο “Εισαγωγή και Κίνητρο”. Σε αυτό το κεφάλαιο δόθηκαν κάποιες ενδιαφέρουσες πληροφορίες και στατιστικά για την υπνηλία, εξηγήθηκε τι είναι η τεχνητή νοημοσύνη και αναφέρθηκαν κάποια κρίσιμα ιστορικά στοιχεία από την πορεία της στον χρόνο και τέλος περιγράφηκε τι εννοούμε όταν αναφερόμαστε στον όρο βαθιά μάθηση, δόθηκαν μερικές σημαντικές πληροφορίες για αυτήν και την λειτουργία της, αναφέρθηκαν κάποιες γνωστές αρχιτεκτονικές νευρωνικών δικτύων και μερικές εφαρμογές της βαθιάς μάθησης στον “κόσμο” των υπολογιστών και κλείνοντας δηλώθηκαν κάποια πλεονεκτήματα αλλά και μειονεκτήματα της έναντι της κλασσικής μηχανικής μάθησης. **Μερικά λόγια για την εξέλιξη της πτυχιακής και τα κεφάλαια που θα ακολουθήσουν μπορούν να βρεθούν παρακάτω.**

2. Προχωρημένες Τεχνικές Μηχανικής Μάθησης:

Σε αυτό το κεφάλαιο θα γίνει μια πιο αναλυτική περιγραφή της θεωρίας των τεχνητών νευρωνικών δικτύων και των αλγορίθμων-συναρτήσεων που σχετίζονται με αυτά, πως λειτουργούν τα δίκτυα, θα δοθεί και μια παρουσίαση για κάποιες γνωστές και χρήσιμες αρχιτεκτονικές αυτών. Τέλος θα περιγραφούν σύντομα κάποιες βιβλιοθήκες κώδικα που χρησιμοποιούνται για την κατασκευή, εκπαίδευση και αξιολόγηση τεχνητών νευρωνικών δικτύων στην γλώσσα προγραμματισμού Python.

3. Εφαρμογές Μηχανικής Μάθησης για Αναγνώριση Υπνηλίας:

Σε αυτό το κεφάλαιο θα παρουσιαστούν οι τρεις κατηγορίες στις οποίες ταξινομούνται τα συστήματα αναγνώρισης υπνηλίας. Θα πραγματοποιηθεί επίσης μια βιβλιογραφική ανασκόπηση κάποιων μεθόδων, τόσο κλασσικής μηχανικής όσο και βαθιάς μάθησης, που χρησιμοποιούνται / έχουν χρησιμοποιηθεί από την επιστημονική κοινότητα για την αναγνώριση της υπνηλίας κατά την οδήγηση. Ακόμα θα ειπωθούν λίγα λόγια για τα πιο γνωστά δημόσια σύνολα δεδομένων που χρησιμοποιούνται στο συγκεκριμένο πεδίο και θα δοθεί η τοποθέτηση της παρούσας εργασίας σε σχέση με την βιβλιογραφία.

4. Το Προτεινόμενο Σύστημα:

Σε αυτό το κεφάλαιο θα περιγραφεί η συνολική αρχιτεκτονική του συστήματος που προτείνει αυτή η εργασία, πως υλοποιήθηκε από την αρχή ως το τέλος, ποια ήταν τα βήματα, ποιες οι τιμές των παραμέτρων που επιλέχθηκαν, ποιες προχωρημένες τεχνικές μηχανικής μάθησης και επεξεργασίας δεδομένων χρησιμοποιήθηκαν για την εξυπηρέτηση των απαιτήσεων-ζητούμενων της εργασίας και γιατί χρησιμοποιήθηκαν.

5. Πειραματική Αποτίμηση:

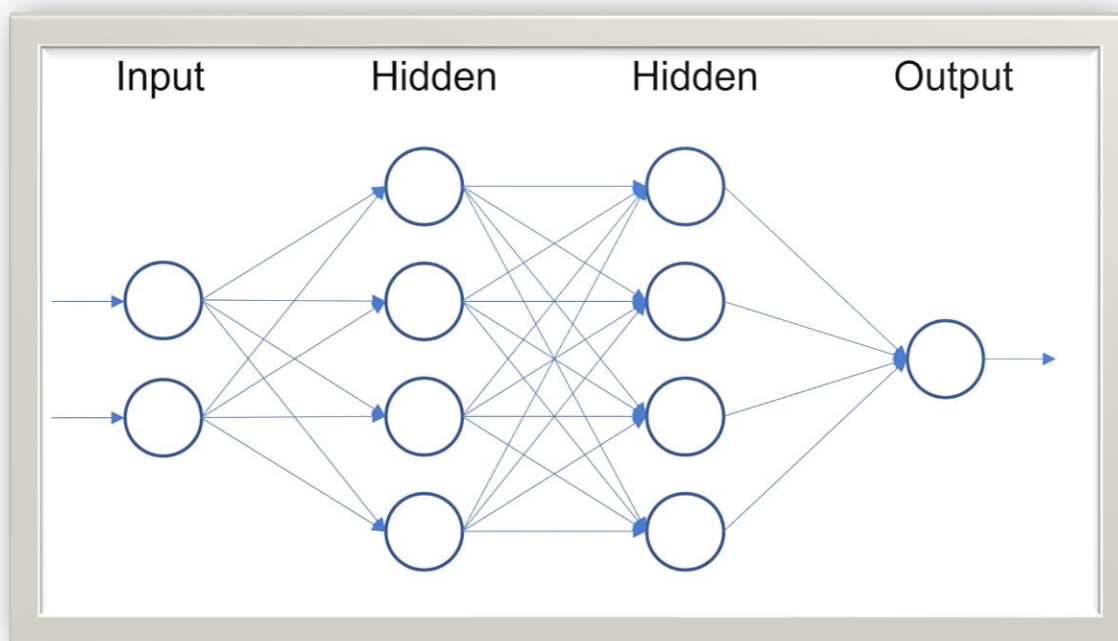
Σε αυτό το κεφάλαιο θα πραγματοποιηθεί μια γενική-συνολική παρουσίαση του πειραματικού περιβάλλοντος και της απόδοσης του μοντέλου σε ότι αφορά την αναγνώριση της υπνηλίας, τι μετρικές χρησιμοποιήθηκαν για την αξιολόγηση του και τι τιμές πήραν αυτές κατά την δοκιμασία του σε δεδομένα που δεν είχε “ξαναδεί”. Ακόμα θα περιγραφεί και ένα *live demo* που δημιουργήθηκε για την “εξέταση” του μοντέλου.

6. Συμπεράσματα και Μελλοντικές Προεκτάσεις:

Σε αυτό το κεφάλαιο θα γίνει μια μικρή παρουσίαση των συμπερασμάτων που προκύπτουν για το σύστημα που προτείνεται, ποια είναι τα πλεονεκτήματα του και ποια τα μειονεκτήματα-περιορισμοί του. Ακόμα θα αναφερθούν τυχόν μελλοντικές προεκτάσεις που θα εξέλισαν την λειτουργία του συστήματος.

2. Προχωρημένες Τεχνικές Μηχανικής Μάθησης

2.1: Πιο αναλυτική εισαγωγή στα τεχνητά νευρωνικά δίκτυα



Εικόνα 2: Δομή ενός Τεχνητού Νευρωνικού Δικτύου

Όπως έχει αναφερθεί ήδη, τα τεχνητά νευρωνικά δίκτυα είναι αριθμητικά/μαθηματικά μοντέλα που προσπαθούν να μιμηθούν την βιολογική λειτουργία του εγκεφάλου με απώτερο στόχο να πλησιάσουν την ανθρώπινη νοημοσύνη και να μπορούν να εκτελούν αποτελεσματικά διάφορες διεργασίες. Τα δίκτυα αυτά τα οποία έχουν δομή σαν αυτήν της εικόνας 2 που φαίνεται παραπάνω αποτελούνται από διάφορες συνιστώσες. Για αρχή δομικό στοιχείο όλων των τεχνητών νευρωνικών δικτύων ανεξάρτητα από την αρχιτεκτονική τους αποτελεί ο τεχνητός νευρώνας, ο οποίος δέχεται εισόδους, “κατέχει” μια αριθμητική τιμή που λέγεται *bias*, έχει μια συνάρτηση ενεργοποίησης και παράγει μια αριθμητική τιμή ως έξοδο. Η αριθμητική τιμή *bias* μετατοπίζει την έξοδο της συνάρτησης ενεργοποίησης δεξιά ή αριστερά ενώ δίνει την δυνατότητα σε έναν νευρώνα να εξαγάγει μια τιμή στην έξοδο του ακόμα και αν όλες οι εισοδοί που πήρε ήταν μηδενικές. Οι νευρώνες συνδέονται μεταξύ τους και ανταλλάσσουν πληροφορία με συνδέσεις οι οποίες χαρακτηρίζονται από αριθμητικές τιμές οι οποίες λέγονται βάρη (*weights*). Τα βάρη καθορίζουν την δύναμη/ισχύ των συνδέσεων. Ακόμα ένα τεχνητό νευρωνικό δίκτυο διαθέτει συναρτήσεις διάδοσης με την βοήθεια των οποίων γίνεται η επεξεργασία και η διάδοση της πληροφορίας μέσα στο δίκτυο από την είσοδο του στην έξοδο περνώντας από τα διάφορα στρώματα νευρώνων της αρχιτεκτονικής του. Τέλος ένα τέτοιο δίκτυο διαθέτει έναν κανόνα μάθησης ο οποίος πρακτικά είναι μια μέθοδος κατά την οποία τροποποιούνται οι τιμές των βαρών

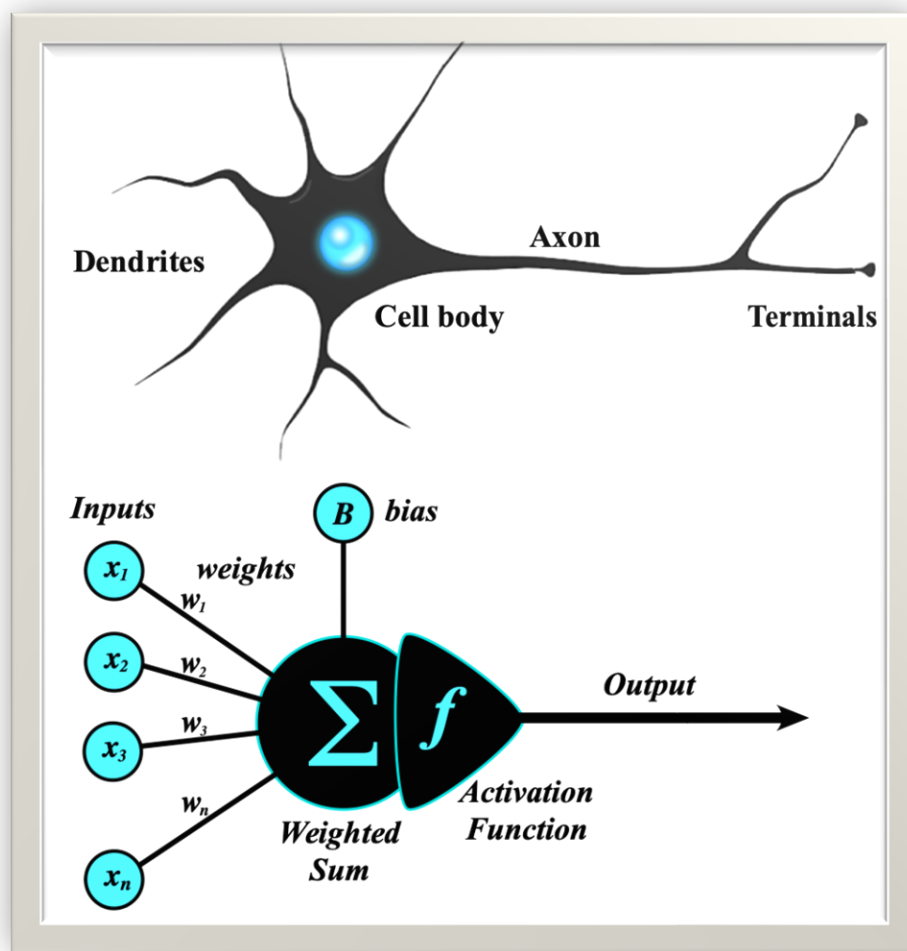
και των *biases* του δικτύου με τελικό στόχο την αύξηση της ακρίβειας του. Περισσότερες πληροφορίες για όλα αυτά τα στοιχεία, την λειτουργία των νευρώνων και κατά συνέπεια του τεχνητού νευρωνικού δικτύου θα δοθούν στο επόμενο κεφάλαιο. Τώρα κοιτώντας την αρχιτεκτονική ενός τεχνητού νευρωνικού δικτύου όπως φαίνεται στο παράδειγμα της Εικόνας 2, παρατηρούμε ότι αποτελείται από 3 είδη επιπέδων/στρωμάτων τεχνητών νευρώνων. Πρώτο στην αρχιτεκτονική συναντάται το επίπεδο εισόδου, σε αυτό το επίπεδο λαμβάνει όλα τα δεδομένα εισόδου το δίκτυο και αποτελείται από τόσους νευρώνες όσα είναι τα διαφορετικά χαρακτηριστικά στα δεδομένα εισόδου, ένας τεχνητός νευρώνας για κάθε χαρακτηριστικό. Αμέσως μετά από αυτό το επίπεδο υπάρχουν τα κρυφά επίπεδα. Τα δεδομένα εισόδου από το επίπεδο εισόδου λαμβάνονται από το πρώτο σε σειρά κρυφό επίπεδο. Ένα δίκτυο πρέπει να έχει τουλάχιστον ένα τέτοιο επίπεδο αλλά μπορεί να έχει και πολλά, αναλόγως με τις απαιτήσεις του προβλήματος/διεργασίας που θέλουμε το δίκτυο να είναι ικανό να επιλύει/υλοποιεί. Όλη η “βαριά” υπολογιστική διαδικασία λαμβάνει χώρα σε αυτά τα επίπεδα τα οποία μετατρέπουν τα δεδομένα εισόδου σε μια μορφή που θα είναι χρήσιμη/συμβατή με το τελευταίο στην σειρά επίπεδο της αρχιτεκτονικής, αυτό της εξόδου. Όπως υπονοεί και το όνομα του, το συγκεκριμένο επίπεδο παράγει την τελική έξοδο του μοντέλου η οποία σε κάθε περίπτωση είναι αριθμητική τιμή, η μορφή της όμως μπορεί να διαφέρει αναλόγως με την διεργασία που υλοποιεί το δίκτυο, κατηγοριοποίηση δεδομένων, παλινδρόμηση κ.α.

2.2: Λειτουργία των τεχνητών νευρωνικών δικτύων

2.2.1: Εμπρός διάδοση (Forward propagation)

Όπως έχει αναφερθεί ήδη, δομικό στοιχείο των τεχνητών νευρωνικών δικτύων αποτελεί ο τεχνητός νευρώνας. Σε αυτό το κεφάλαιο θα εξηγηθεί αναλυτικότερα η λειτουργία του και κατά συνέπεια η λειτουργία όλου του δικτύου με την τεχνική της εμπρός διάδοσης (*Forward Propagation*), η οποία χρησιμοποιείται τόσο κατά την “εκπαίδευση” του όσο και κατά την αξιολόγηση αλλά και μετέπειτα εφαρμογή του. Με αυτήν την τεχνική τα δεδομένα που εισέρχονται στο δίκτυο στο επίπεδο εισόδου περνάνε από όλα τα κρυφά επίπεδα, στα οποία πραγματοποιείται η συντριπτική πλειονότητα των πράξεων που απαιτούνται, για να καταλήξουν τελικά μορφοποιημένα στο επίπεδο εξόδου με σκοπό να παραχθεί η τελική έξοδος του μοντέλου.

Η δομή ενός τεχνητού νευρώνα καθώς και ενός βιολογικού, που συναντάται στα νευρωνικά δίκτυα του εγκεφάλου, από τον οποίο αντλεί έμπνευση τόσο για την δομή του όσο και για την λειτουργία του ο τεχνητός, φαίνεται στην εικόνα που ακολουθεί.



Εικόνα 3: Βιολογικός και Τεχνητός Νευρώνας

Κατά την διαδικασία λοιπόν της εμπρός διάδοσης, κάθε τεχνητός νευρώνας σε κάθε εσωτερικό επίπεδο του τεχνητού νευρωνικού δικτύου δέχεται εισόδους είτε από το επίπεδο εισόδου, άρα τα δεδομένα εισόδου, είτε από άλλους ενδιάμεσους νευρώνες στα κρυφά επίπεδα. Ο κάθε νευρώνας είναι πρακτικά μια συνάρτηση η οποία πολλαπλασιάζει κάθε είσοδο που δέχεται με την αριθμητική τιμή του βάρους που σχετίζεται με την σύνδεση της. Έπειτα όλα αυτά τα γινόμενα αθροίζονται μεταξύ τους και προστίθεται στο αποτέλεσμα η αριθμητική τιμή bias, η οποία είναι ξεχωριστή για κάθε τεχνητό νευρώνα. Η διαδικασία αυτή μπορεί να αναπαρασταθεί μαθηματικά ως εξής:

$$z = \sum_{i=1}^n (x_i \cdot w_i) + b$$

Εξίσωση 1: Μαθηματική Πράξη TN

Όπου η ερμηνεία των συμβόλων που συμμετέχουν στην παραπάνω μαθηματική σχέση δίνεται παρακάτω:

- **Σ :** Μαθηματικό σύμβολο άθροισης
- **z :** Η έξοδος του νευρώνα
- **n :** Το πλήθος των συνολικών εισόδων που δέχτηκε ο νευρώνας
- **x_i :** Η i -οστή είσοδος στον νευρώνα
- **w_i :** Το i -οστό βάρος του νευρώνα
- **b :** Η αριθμητική τιμή bias

Αφού λοιπόν γίνει αυτός ο υπολογισμός το αποτέλεσμα της πράξης δίνεται ως είσοδος σε μια ακόμα συνάρτηση, την συνάρτηση ενεργοποίησης, η οποία παράγει και την τελική έξοδο του νευρώνα με βάση τον τύπο της. Η συνάρτηση αυτή εμφανίζεται μετά από κάθε νευρώνα εκτός από αυτούς που ανήκουν στο επίπεδο εισόδου ίσως και από αυτούς που ανήκουν στο τελευταίο επίπεδο, αυτό της εξόδου. Η συνάρτηση αυτή είναι πολύ κρίσιμη για την λειτουργία ενός τεχνητού νευρωνικού δικτύου καθώς εισάγει μη γραμμικότητα στο δίκτυο. Χωρίς αυτήν το μοντέλο συμπεριφέρεται ως ένα γραμμικό μοντέλο πρόβλεψης, όπως είναι π.χ. ο γνωστός αλγόριθμος της γραμμικής παλινδρόμησης (*Linear Regression*) για την πρόβλεψη συνεχόμενων τιμών, κάτι το οποίο δεν είναι θεμιτό καθώς τα δεδομένα από τον πραγματικό κόσμο με τα οποία και “εκπαιδεύεται” το δίκτυο οπάνια είναι γραμμικά διαχωριζόμενα, τις περισσότερες φορές εμπεριέχεται και το στοιχείο του θορύβου. Εντάσσοντας την συνάρτηση ενεργοποίησης στο δίκτυο πρακτικά του δίνεται η δυνατότητα να “μάθει” πιο σύνθετα μοτίβα στα δεδομένα εισόδου αυξάνοντας κατά συνέπεια και την ακρίβεια/απόδοση του. Ανάμεσα στις πολλές διαθέσιμες συναρτήσεις ενεργοποίησης που μας προσφέρουν τα μαθηματικά στις πιο γνωστές ανήκουν η *relu* (*Rectified Liner Unit*) και η *sigmoid*. Οι μαθηματικοί τύποι αυτών όπως και οι γραφικές τους αναπαραστάσεις, καθώς και λίγα λόγια για την λειτουργία τους δίνονται στην συνέχεια:

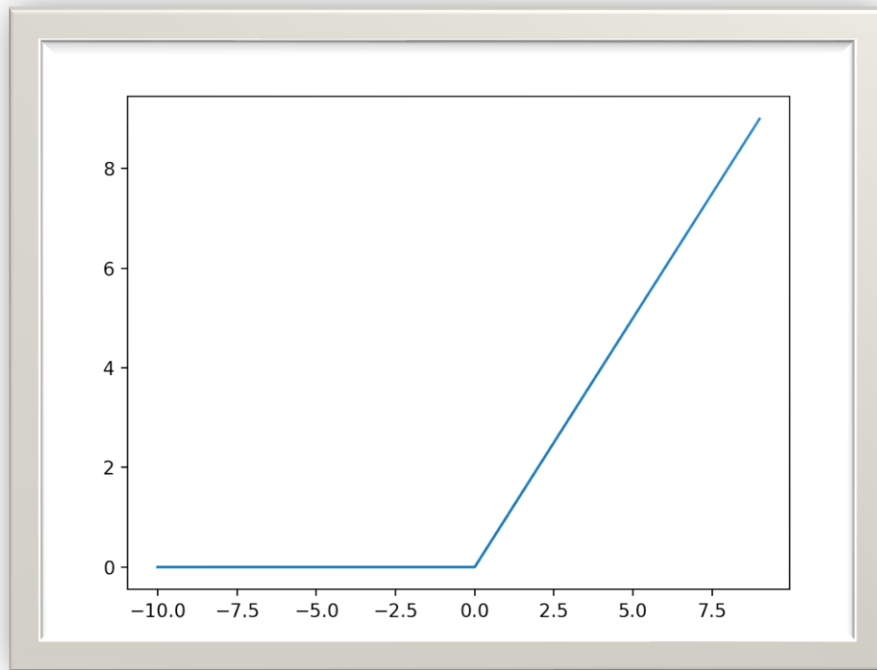
Relu:

$$f(x) = \max(0, x)$$

Εξίσωση 2: Συναρτησιακός Τύπος Relu

Επί της ουσίας η *relu* επιστρέφει στην έξοδο την είσοδο που δέχτηκε αν αυτή ήταν θετική αλλιώς αν ήταν αρνητική επιστρέφει μηδέν. Λόγω του πολύ απλού υπολογισμού που υλοποιεί είναι υπολογιστικά αποδοτική κάνοντας την μια πολύ διαδεδομένη και ευρέως χρησιμοποιούμενη συνάρτηση ενεργοποίησης, κυρίως για τα κρυφά επίπεδα του τεχνητού νευρωνικού δικτύου τα οποία διαχειρίζονται και το μεγαλύτερο “βάρος” των υπολογισμών, αυξάνοντας τελικά την ταχύτητα της εκπαίδευσης και την απόδοση/ακρίβεια του δικτύου. Ακόμα συμβάλλει και στην μείωση του *Vanishing*

Gradient φαινομένου, το οποίο επί της ουσίας δεν επιτρέπει στο δίκτυο να εκπαιδευτεί σωστά, από το οποίο “πάσχουν” τα δίκτυα που αποτελούνται από πολλά επίπεδα τεχνητών νευρώνων ή χρησιμοποιούν άλλες συναρτήσεις ενεργοποίησης όπως π.χ. η tanh. Παρακάτω ακολουθεί η γραφική παράσταση της συγκεκριμένης συνάρτησης ενεργοποίησης.



Εικόνα 4: Γραφική Αναπαράσταση της Relu

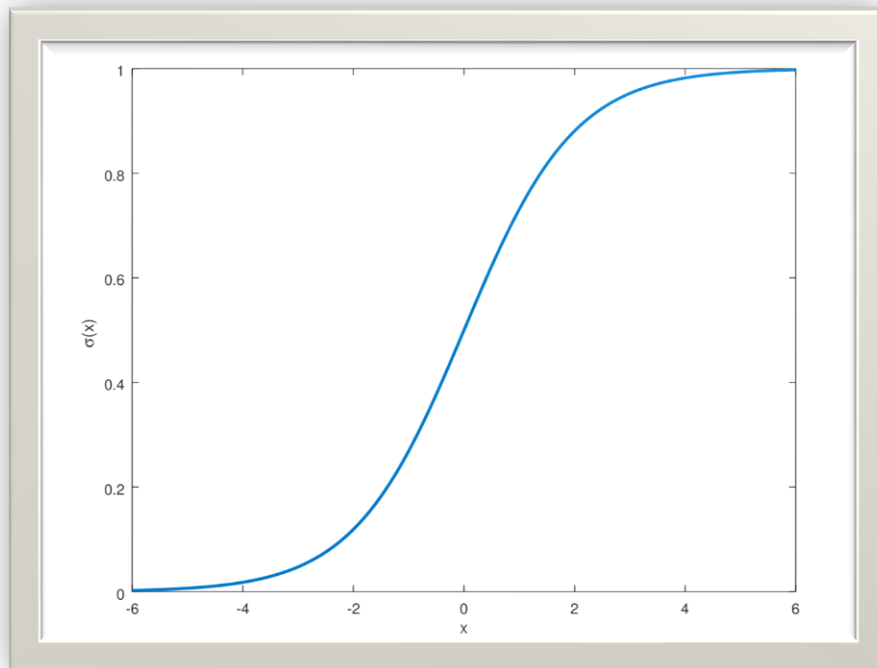
Sigmoid:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Εξίσωση 3: Συναρτησιακός Τύπος Sigmoid

Η συγκεκριμένη συνάρτηση διασφαλίζει ομαλή και συνεχή έξοδο η οποία είναι απαραίτητη για μεθόδους βελτιστοποίησης που βασίζονται στις κλίσεις (gradients) των βαρών και των biases. Επειδή η έξοδος της, για οποιαδήποτε είσοδο, είναι πάντα στο διάστημα (0,1) την καθιστά χρήσιμη επιλογή για προβλήματα δυαδικής ταξινόμησης, άρα είναι μια συνάρτηση ενεργοποίησης που χρησιμοποιείται κυρίως στο επίπεδο εξόδου ενός τεχνητού νευρωνικού δικτύου. Όπως φαίνεται και από την εικόνα με την γραφική της παράσταση που ακολουθεί, η συνάρτηση παρουσιάζει απότομη κλίση όταν οι τιμές εισόδου είναι ανάμεσα στο [-2, 2] με αποτέλεσμα μικρές αλλαγές στις τιμές εισόδου που ανήκουν σε αυτό το διάστημα να προκαλούν σημαντικές αλλαγές

στην έξοδο της κάτι το οποίο είναι κρίσιμο για την εκπαίδευση ενός τεχνητού νευρωνικού δικτύου.



Εικόνα 5: Γραφική Παράσταση της Sigmoid

Τελειώνοντας την περιγραφή και των συναρτήσεων ενεργοποίησης πρακτικά ο κύκλος της εμπρός διάδοσης των δεδομένων σε ένα τεχνητό νευρωνικό δίκτυο κλείνει, στην συνέχεια ακολουθεί η διαδικασία της πίσω διάδοσης του σφάλματος και της τροποποίησης των βαρών και των *biases* από έναν αλγόριθμο που ονομάζεται βελτιστοποιητής. Περισσότερες πληροφορίες στο κεφάλαιο που ακολουθεί.

2.2.2: Πίσω διάδοση (Backpropagation)

Αφού λοιπόν ολοκληρωθεί η διαδικασία της εμπρόσθιας διάδοσης που περιγράφηκε νωρίτερα στο προηγούμενο κεφάλαιο ξεκινάει η διαδικασία της προς τα πίσω διάδοσης. Στο πρώτο βήμα αυτής της διαδικασίας το μοντέλο αξιολογεί την απόδοση του, στην διεργασία που του έχει “ανατεθεί”, χρησιμοποιώντας μια συνάρτηση απώλειας η οποία ποσοτικοποιεί την διαφορά ανάμεσα στις προβλέψεις του και στις πραγματικές επικέτες των δεδομένων εισόδου, υπολογίζει δηλαδή με απλά λόγια το σφάλμα. Η συνάρτηση απώλειας είναι πολύ κρίσιμη για την σωστή λειτουργία του δικτύου όπου στόχος του είναι στο τέλος της “εκπαίδευσης” να έχει μειώσει όσο το δυνατόν περισσότερο τον αριθμό που εξάγει η συνάρτηση αυτή. Υπάρχουν πολλές διαφορετικές συναρτήσεις απώλειας που μπορούν να χρησιμοποιηθούν, η επιλογή εξαρτάται κυρίως από την διεργασία που θέλουμε το δίκτυο να υλοποιεί, π.χ. η

μαθηματική συνάρτηση απώλειας *Mean Squared Error (MSE)* είναι μια κοινή επιλογή για εφαρμογές παλινδρόμησης (θέλουμε δηλαδή το δίκτυο να προβλέπει συνεχείς τιμές π.χ. θερμοκρασία) ενώ η *cross entropy* χρησιμοποιείται ευρέως για εφαρμογές κατηγοριοποίησης. Παρακάτω δίνονται οι μαθηματικοί τύποι των προαναφερθέντων συναρτήσεων καθώς και λίγες παραπάνω πληροφορίες για αυτές:

Mean Squared Error (MSE):

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

Εξίσωση 4: Μαθηματικός Τύπος MSE

Όπου η περιγραφή των συμβόλων είναι η ακόλουθη:

- **n :** Το πλήθος των προβλέψεων του μοντέλου
- **y_i :** Η πραγματική τιμή για το i -οστό δείγμα δεδομένων από το σύνολο αυτών
- **$\hat{y}_i$:** Η προβλεπόμενη τιμή για το i -οστό δείγμα δεδομένων από το σύνολο αυτών

Η συγκεκριμένη συνάρτηση επί της ουσίας υψώνει κάθε διαφορά ανάμεσα στην πραγματική και την προβλεπόμενη τιμή στο τετράγωνο, αθροίζει όλους αυτούς τους αριθμούς μαζί και στο τέλος διαιρεί το αποτέλεσμα με το πλήθος των προβλέψεων ώστε να βρει τον μέσο όρο αυτών. Στα τεχνητά νευρωνικά δίκτυα η ύψωση των διαφορών στο τετράγωνο είναι ιδιαίτερα σημαντική και κρίσιμη καθώς με αυτόν τον τρόπο διαχειρίζονται καλύτερα και οι αρνητικές αλλά και οι θετικές τιμές/σφάλματα. Σε ένα μοντέλο οι διαφορές αυτές, τα σφάλματα δηλαδή, μπορούν να είναι τόσο αρνητικοί αλλά και θετικοί αριθμοί, η άθροιση αριθμών διαφορετικού προσήμου μπορεί να οδηγήσει σε ένα συνολικό σφάλμα ίσο με το μηδέν κάτι το οποίο είναι λανθασμένο και προβληματικό καθώς θα “οδηγούσε” το τεχνητό νευρωνικό δίκτυο να μην προχωρήσει σε προσαρμογές των βαρών και των *biases* του νομίζοντας πως τα πηγαίνει εξαιρετικά ακόμα και αν δεν το κάνει. Το πρόβλημα αυτό επιλύεται με την ύψωση στο τετράγωνο. Επιπλέον η διαδικασία της ύψωσης στο τετράγωνο προσδίδει περισσότερο “βάρος” σε υψηλές τιμές σφάλματος έναντι χαμηλότερων οδηγώντας το μοντέλο να πραγματοποιήσει μεγαλύτερες αλλαγές στα βάρη και τα *biases* του με σκοπό να μειώσει αυτήν την τιμή του σφάλματος, βελτιώνοντας κατά συνέπεια την απόδοση του πιο γρήγορα. Από την άλλη όμως η ύψωση στο τετράγωνο καθιστά την συνάρτηση αυτήν “ευαίσθητη” σε ακραίες τιμές (*outliers*), οι οποίες εξαιτίας του υψηλού σφάλματος που θα παράξουν έχουν σημαντική επίδραση στην εκπαίδευση του τεχνητού νευρωνικού δικτύου. Τέλος η συνάρτηση αυτή είναι ανεξάρτητη από το πλήθος δειγμάτων του αρχικού συνόλου δεδομένων εκπαίδευσης, λόγω της διαίρεσης που πραγματοποιεί με το πλήθος των προβλέψεων στο άθροισμα του τύπου της

υπολογίζοντας τον μέσο όρο, καθιστώντας την μια σταθερή και αξιόπιστη επιλογή για χρήση ανάμεσα σε σύνολα δεδομένων διαφορετικού μεγέθους [4].

Cross Entropy Loss:

$$\text{CrossEntropyLoss} = - \sum_{i=1}^n y_i \log \hat{y}_i$$

Εξίσωση 5: Μαθηματικός Τύπος CEL

Η cross entropy loss είναι μια συνάρτηση απώλειας που έχει γίνει ιδιαίτερα δημοφιλής και χρήσιμη για προβλήματα ταξινόμησης στην μηχανική μάθηση τα οποία μπορούν να οριστούν ως την προσπάθεια το τεχνητό νευρωνικό δίκτυο να είναι σε θέση να βρει την πραγματική κλάση/ετικέτα στην οποία ανήκει ένα δείγμα δεδομένων, π.χ. μια εικόνα, από ένα σύνολο υπαρκτών ετικετών/κλάσεων στο σύνολο δεδομένων εκπαίδευσης. Η συγκεκριμένη συνάρτηση “επιβραβεύει” το μοντέλο, παράγεται δηλαδή μικρή αριθμητική τιμή σφάλματος, όταν εξάγει υψηλές πιθανότητες για τις σωστές απαντήσεις (η προβλεπόμενη ετικέτα ταιριάζει με την πραγματική) σε προβλήματα κατηγοριοποίησης ενώ “τιμωρεί” το μοντέλο, παράγεται δηλαδή μεγάλο σφάλμα, όταν το μοντέλο εξάγει υψηλές πιθανότητες για λανθασμένες απαντήσεις. Ακόμα εξαιτίας της λογαριθμικής της φύσης είναι “ευαίσθητη”, η τιμή δηλαδή του σφάλματος που παράγει μεταβάλλεται περισσότερο και ευκολότερα, σε αλλαγές σε αβέβαιες πιθανότητες του μοντέλου για τις σωστές κλάσεις. Γενικά η συγκεκριμένη συνάρτηση απώλειας “ωθεί” το μοντέλο να επιλύσει την αβεβαιότητα στις προβλέψεις και να εξάγει υψηλές πιθανότητες για τις σωστές κλάσεις [5].

Έπειτα από τον υπολογισμό της τιμής σφάλματος, στην διαδικασία της προς τα πίσω διάδοσης ακολουθεί η διαδικασία του υπολογισμού των κλίσεων του τεχνητού νευρωνικού δικτύου. Ο αλγόριθμος που αναλαμβάνει αυτό το έργο ονομάζεται *Backpropagation*. Οι κλίσεις που υπολογίζονται από αυτόν τον αλγόριθμο είναι οι μερικές παράγωγοι της συνάρτησης σφάλματος ως προς κάθε βάρος και bias του δικτύου. Για να υπολογιστούν χρησιμοποιείται η τεχνική της αλυσίδας από τον απειροστικό λογισμό των μαθηματικών με σκοπό να προσδιοριστεί ο βαθμός κατά τον οποίο κάθε τμήμα του σφάλματος εξόδου οφείλεται σε κάθε βάρος και bias. Οι κλίσεις αυτές είναι απαραίτητες κατά την διαδικασία προσαρμογής των βαρών και των biases, των εκπαιδευσιμων δηλαδή παραμέτρων του δικτύου, με στόχο την μείωση της τιμής της συνάρτησης απώλειας. Επομένως μετά τον υπολογισμό των κλίσεων από τον *Backpropagation* αλγόριθμο ένας άλλος αλγόριθμος, ο βελτιστοποιητής (*Optimizer*), λαμβάνει αυτές τις κλίσεις και πραγματοποιεί την προαναφερθείσα τροποποίηση των εκπαιδευσιμων παραμέτρων. Πρακτικά τροποποιεί την τιμή κάθε βάρους και bias προς την αντίθεση κατεύθυνση της κλίσης (αφαιρεί ή προσθέτει μια τιμή σε αυτό) ενώ το μέγεθος της τροποποίησης που συμβαίνει (το μέγεθος της τιμής που αφαιρείται/προστίθεται) σε κάθε ενημέρωση που πραγματοποιεί μέχρι το τέλος της εκπαίδευσης εξαρτάται άμεσα από μια υπερπαραμέτρο (*Hyperparameter*) που

ονομάζεται ρυθμός μάθησης (*Learning Rate*) και ορίζεται από τον προγραμματιστή πριν ξεκινήσει η διαδικασία της εκπαίδευσης του τεχνητού νευρωνικού δικτύου. Υπάρχουν πολλοί διαθέσιμοι αλγόριθμοι που υλοποιούν αυτήν την διαδικασία, ο καθένας με τις δικές του τεχνικές, ιδιαιτερότητες και αδυναμίες, η επιλογή του σωστού αλγορίθμου εξαρτάται από το εκάστοτε πρόβλημα/διεργασία, από τα χαρακτηριστικά του συνόλου δεδομένων εκπαίδευσης και από τους υπολογιστικούς πόρους που είναι διαθέσιμοι στον υπολογιστή, δύο από τους πιο σημαντικούς και ευρέως χρησιμοποιούμενους πλέον είναι ο *Adam* και ο *RMSProp*. Μερικά λόγια για την λειτουργία τους καθώς και κάποια πλεονεκτήματα αλλά και μειονεκτήματα τους δίνονται στην συνέχεια:

Adam:

Adaptive Moment Estimation, ο πιο διαδεδομένος και ευρέως χρησιμοποιούμενος πλέον αλγόριθμος βελτιστοποίησης, αποτελεί την βασική επιλογή για τα περισσότερα προβλήματα βαθιάς μάθησης. Ο συγκεκριμένος αλγόριθμος εκμεταλλεύεται τα πλεονεκτήματα των *Momentum* και *RMSProp* προσφέροντας του στιβαρή απόδοση στις περισσότερες περιπτώσεις. Διατηρεί δυναμικό ρυθμό μάθησης (*learning rate*) για κάθε βάρος και *bias* ξεχωριστά χρησιμοποιώντας την πρώτη ροπή (μέση τιμή) και την δεύτερη ροπή (διακύμανση) των κλίσεων αυτών. Με βάση τον δυναμικό ρυθμό μάθησης και την μέση τιμή των κλίσεων, τροποποιεί την τιμή κάθε εκπαιδευσιμής παραμέτρου. Το βασικό πλεονέκτημα του είναι ότι προκαλεί γρήγορη σύγκλιση στο τεχνητό νευρωνικό δίκτυο δηλαδή σταθεροποιούνται οι εκπαιδευσιμες παράμετροι και οποιαδήποτε επιπλέον αλλαγή σε αυτές δεν προκαλεί αισθητή μείωση στο σφάλμα του μοντέλου. Από την άλλη όμως είναι υπολογιστικά απαιτητικός και έχει ανάγκη δέσμευσης σημαντικής ποσότητας μνήμης του υπολογιστή κατά την εκτέλεση καθώς πρέπει να αποθηκεύει την πρώτη και την δεύτερη ροπή.

RMSProp:

RMSProp, ένας ακόμα ιδιαίτερα δημοφιλής αλγόριθμος βελτιστοποίησης ο οποίος υιοθετεί δυναμικό ρυθμό μάθησης για κάθε εκπαιδευσιμη παράμετρο ξεχωριστά. Η έμπνευση για την δημιουργία του προήλθε από τους περιορισμούς αλγορίθμων που είχαν προταθεί πριν από αυτόν. Βασικό του χαρακτηριστικό είναι πως μειώνει το μέγεθος της τροποποίησης της τιμής εκπαιδευσιμων παραμέτρων που παρουσιάζουν έντονες ταλαντώσεις, έντονες αυξομειώσεις στις κλίσεις τους, ελαττώνοντας έτσι την αστάθεια κατά την εκπαίδευση. Ένα από τα βασικά του πλεονεκτήματα είναι πως αποτελεί μια αξιόπιστη επιλογή όταν το σύνολο δεδομένων εκπαίδευσης είναι μεγάλο και το τροφοδοτούμε στο τεχνητό νευρωνικό δίκτυο σε παρτίδες (*mini batches*). Επιπλέον χρειάζεται λιγότερο *tuning*, πείραγμα δηλαδή των υπερπαραμέτρων του με σκοπό να βρεθούν οι ιδανικές τιμές για το εκάστοτε πρόβλημα, σε σχέση με άλλους αλγορίθμους, είναι ιδιαίτερα αποτελεσματικός και στιβαρός σε προβλήματα ενισχυτικής μάθησης όπου το τι πρέπει να κάνει το δίκτυο αλλάζει δυναμικά με την πάροδο του χρόνου κ.α. Από την άλλη στα μειονεκτήματα του συγκαταλέγονται η πιθανότητα σε κάποιες περιπτώσεις να οδηγήσει σε υπερπροσαρμογή το δίκτυο, το

δίκτυο μαθαίνει πολύ καλά τα δεδομένα εκπαίδευσης απομνημονεύοντας τον θόρυβο σε αυτά χωρίς να είναι σε θέση να αναγνωρίζει τα γενικά μοτίβα αποτελεσματικά με συνέπεια η απόδοση του σε νέα δεδομένα να είναι χαμηλή, όπως και η ανάγκη να οριστεί από τον προγραμματιστή η καθολική τιμή που χρησιμοποιεί του ρυθμού μάθησης όπου αρκετές φορές το να βρεθεί η ιδανική τιμή για το εκάστοτε πρόβλημα είναι απαιτητικό και χρονοβόρο [6].

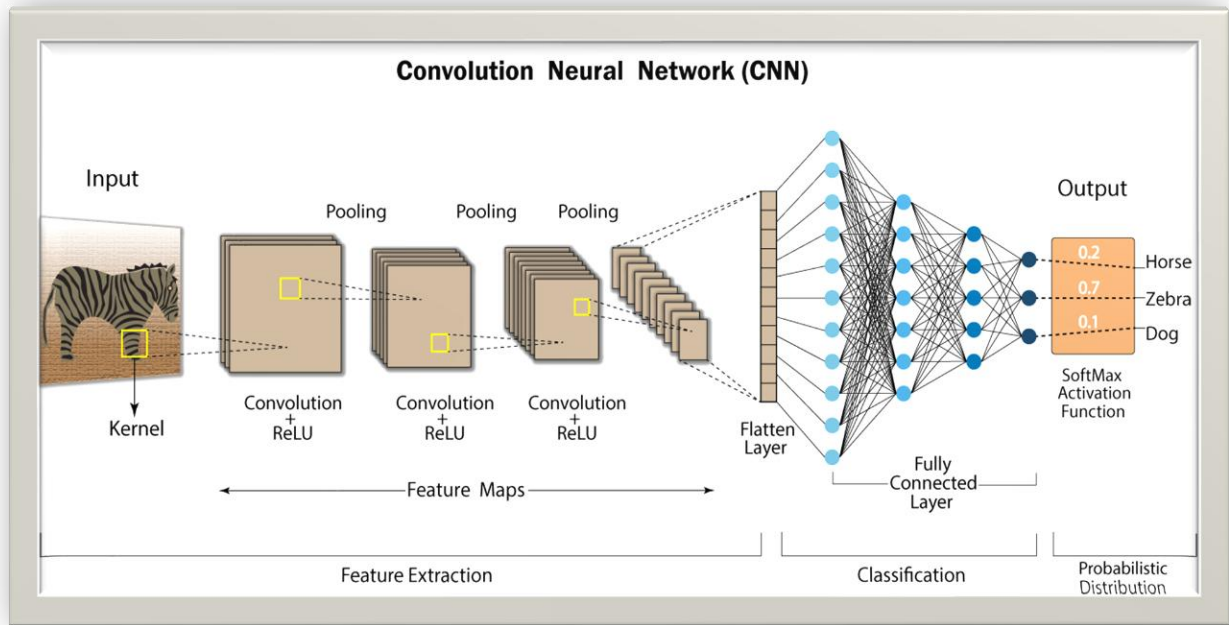
2.2.3: Επαναληπτική διαδικασία

Η διαδικασία που περιγράφηκε στα προηγούμενα κεφάλαια δηλαδή η εμπρός διάδοση των δεδομένων εκπαίδευσης, ο υπολογισμός του σφάλματος ανάμεσα στις προβλέψεις του τεχνητού νευρωνικού δικτύου και στις πραγματικές ετικέτες/κλάσεις των δειγμάτων από το σύνολο των δεδομένων, ο υπολογισμός των κλίσεων με βάση τον αλγόριθμο της προς τα πίσω διάδοσης (Backpropagation) και η τροποποίηση των εκπαιδευσιμων παραμέτρων του δικτύου (βαρών και biases) με βάση τον αλγόριθμο βελτιστοποίησης (Optimizer) δεν πραγματοποιείται μόνο μια φορά “πάνω” στο σύνολο των δεδομένων εκπαίδευσης αλλά είναι μια επαναληπτική διαδικασία που συμβαίνει έως ότου το μοντέλο να φτάσει σε σύγκλιση. Σκοπός της διαδικασίας αυτής είναι μετά από κάθε επανάληψη το σφάλμα του δικτύου να μειώνεται, κατά συνέπεια οι προβλέψεις του να βελτιώνονται, έως ότου να μην μπορεί να μειωθεί άλλο αισθητά.

2.3: Συνελικτικά νευρωνικά δίκτυα (CNN)

Τα συνελικτικά νευρωνικά δίκτυα είναι μια πολύ δημοφιλής και ευρέως χρησιμοποιούμενη αρχιτεκτονική τεχνητών νευρωνικών δικτύων, εμπνευσμένη από το οπτικό σύστημα του ανθρώπου, τα οποία χρησιμοποιούνται για την αναγνώριση ψηφιακών εικόνων βρίσκοντας πολλές εφαρμογές στον τομέα της υπολογιστικής όρασης. Αποτελούνται από μια σειρά επιπέδων, όπως φαίνεται και στην εικόνα που ακολουθεί, με αυτά να είναι το επίπεδο της εισόδου, τα επίπεδα συνέλιξης και αυτά της ενεργοποίησης, τα επίπεδα ομαδοποίησης (Pooling), το επίπεδο ισοπέδωσης (Flatten), το πλήρως συνδεδεμένο επίπεδο και τέλος αυτό της εξόδου. Περνώντας σε λεπτομέρειες το επίπεδο εισόδου δέχεται τους πίνακες με τα δεδομένα των εικόνων και τα προωθεί στο υπόλοιπο δίκτυο για επεξεργασία. Στην συνέχεια ακολουθεί το επίπεδο συνέλιξης, το πιο σημαντικό σε ένα τέτοιο δίκτυο, το οποίο είναι σχεδιασμένο να εξάγει σημαντικά χαρακτηριστικά και μοτίβα από τις εικόνες, από απλές γραμμές στα πρώτα συνελικτικά επίπεδα μέχρι πιο σύνθετα σχήματα και αντικείμενα/μορφές στα τελευταία. Το κάνει αυτό χρησιμοποιώντας φίλτρα, επί της ουσίας μικρούς πίνακες διαστάσεων π.χ. 2×2 , τα οποία περιέχουν εκπαιδευσιμες παραμέτρους (βάρη). Αυτά τα φίλτρα ολισθαίνουν πάνω στις εικόνες με βάση ένα βήμα (stride) π.χ. δύο pixels και υπολογίζεται το εσωτερικό γινόμενο ανάμεσα στα βάρη αυτών και τα pixels των τμημάτων των εικόνων, έπειτα σε κάθε εσωτερικό γινόμενο προστίθεται ένα ξεχωριστό bias, με αποτέλεσμα να παράγονται πίνακες χαρακτηριστικών. Σε κάθε τέτοιο επίπεδο

παράγονται τόσοι πίνακες χαρακτηριστικών όσα ήταν τα φίλτρα που εφαρμόστηκαν. Σημαντικό είναι επίσης να αναφερθεί ότι λόγω της φύσης αυτής της διαδικασίας μπορούν να εντοπίζονται σχήματα και αντικείμενα στις εικόνες ανεξάρτητα από την θέση τους σε αυτές και όπως φαίνεται και από την παρακάτω εικόνα ένα CNN μπορεί και πρέπει περιέχει διαφορετικά επίπεδα συνέλιξης το καθένα με τα δικά του φίλτρα με το κάθε φίλτρο να περιέχει διαφορετικά βάρη.



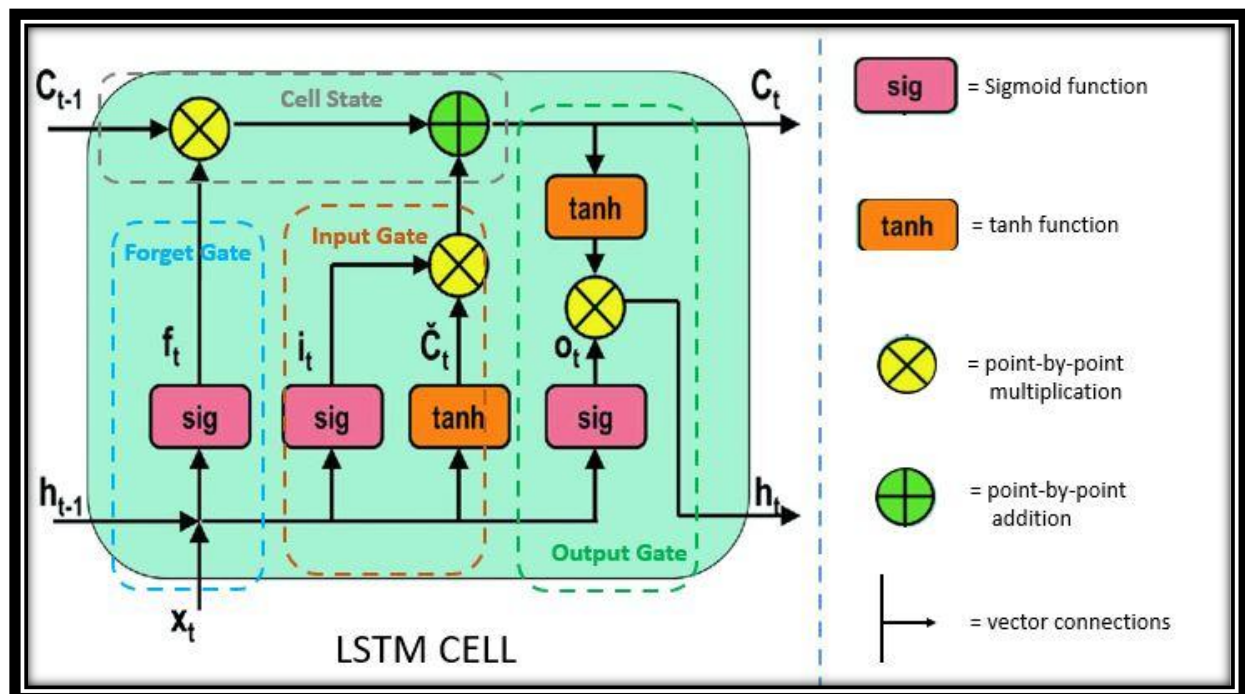
Εικόνα 6: Συνελικτικό Νευρωνικό Δίκτυο

Έπειτα ακολουθεί το επίπεδο της ενεργοποίησης που εφαρμόζει μια συνάρτηση ενεργοποίησης σε κάθε στοιχείο των πινάκων χαρακτηριστικών. Αμέσως μετά ακολουθεί το επίπεδο της ομαδοποίησης (*Pooling*), συνηθίζεται να τοποθετείται ανάμεσα στα επίπεδα συνέλιξης όπως φαίνεται και στην εικόνα, ο ρόλος του οποίου είναι να ελαττώσει τις διαστάσεις των πινάκων χαρακτηριστικών, μόνο το μήκος και το πλάτος όχι το βάθος. Η πιο γνωστή και διαδεδομένη μέθοδος είναι η μέγιστη ομαδοποίηση (*Max Pooling*). Χρησιμοποιεί ένα πλέγμα μεγέθους π.χ. 2×2 , χωρίς εκπαιδευσιμες παραμέτρους, το οποίο ολισθαίνει με βάση ένα βήμα πάνω στον πίνακα χαρακτηριστικών επιλέγοντας κάθε φορά το μέγιστο στοιχείο του τμήματος που “βλέπει”. Η διαδικασία της ομαδοποίησης είναι ιδιαίτερα ωφέλιμη καθώς μειώνει τον αριθμό των μετέπειτα υπολογισμών που θα πρέπει να πραγματοποιηθούν, ελαττώνει την χρήση της μνήμης του υπολογιστή και βοηθάει στην αποφυγή της υπερπροσαρμογής του τεχνητού νευρωνικού δικτύου. Μετά το επίπεδο αυτό ακολουθεί, και όταν πλέον έχει τελειώσει η εξαγωγή χαρακτηριστικών από το δίκτυο, το επίπεδο της ισοπέδωσης (*Flatten*), βλέπε εικόνα. Το επίπεδο αυτό πρακτικά μετατρέπει όλους τους 3D πίνακες χαρακτηριστικών που δέχτηκε στην είσοδο του σε ένα ενιαίο μονοδιάστατο διάνυσμα. Για παράδειγμα αν δέχτηκε είσοδο μεγέθους (16, 16, 32) θα παράξει ένα διάνυσμα μεγέθους (8192,). Το μονοδιάστατο αυτό διάνυσμα εισέρχεται μετά στο πλήρως συνδεδεμένο επίπεδο, αποτελεί είσοδο σε κάθε νευρώνα αυτού, ο

κάθε νευρώνας με την γνωστή διαδικασία που έχει αναφερθεί ήδη υπολογίζει μια τιμή η οποία αποτελεί είσοδο σε κάθε νευρώνα του τελευταίου επιπέδου, αυτό της εξόδου. Τέλος το επίπεδο της εξόδου με την σειρά του μετατρέπει αυτές τις “ωμές” τιμές των προηγούμενων νευρώνων σε πιθανότητες, χρησιμοποιώντας την συνάρτηση ενεργοποίησης *sigmoid* για δυαδική ταξινόμηση ή την *softmax* για ταξινόμηση ανάμεσα σε πάνω από δύο κλάσεις.

2.4: Long short-term memory (LSTM)

Αυτή η αρχιτεκτονική τεχνητών νευρωνικών δικτύων αποτελεί εξέλιξη των κλασσικών αναδρομικών νευρωνικών δικτύων (RNNs) και έχει σχεδιαστεί ώστε να μπορεί να εντοπίζει και να θυμάται εξαρτήσεις μακράς διάρκειας σε συνεχόμενα δεδομένα. Επιλύει τους περιορισμούς και τα μειονεκτήματα των RNNs, βασικό μειονέκτημα τους αποτελεί η αδυναμία τους να θυμούνται εξαρτήσεις μακράς διάρκειας, υιοθετώντας ένα κελί μνήμης (*memory cell*), επί της ουσίας ένα διάνυσμα αριθμών κινητής υποδιαστολής, το οποίο αποθηκεύει σημαντική πληροφορία με την πάροδο του χρόνου. Εκτός από το κελί μνήμης ένα επίπεδο LSTM διαθέτει και τρεις πύλες, επίσης επί της ουσίας διανύσματα αριθμών κινητής υποδιαστολής, οι οποίες είναι, όπως φαίνεται και από την εικόνα που ακολουθεί, αυτή της εισόδου, αυτή της λήθης (*Forget Gate*) και αυτή της εξόδου.



Εικόνα 7: Αρχιτεκτονική LSTM Επιπέδου

Περνώντας σε λεπτομέρειες η πύλη εισόδου επί της ουσίας ελέγχει ποια πληροφορία του εκάστοτε χρονικού βήματος θα εισαχθεί στο κελί μνήμης, η πύλη λήθης ποια πληροφορία θα εξαχθεί από αυτό ως μη χρήσιμη πλέον ενώ η πύλη εξόδου ελέγχει ποια πληροφορία από το κελί μνήμης θα περάσει στην επόμενη κρυφή κατάσταση (*hidden state*), την έξοδο δηλαδή του επιπέδου.

Πύλη Λήθης:

Χρησιμοποιεί την νέα είσοδο (x_t) και το διάνυσμα της προηγούμενης κρυφής κατάστασης $[h_{(t-1)}]$ και εφαρμόζει τον ακόλουθο μαθηματικό τύπο:

$$f_t = \text{sig}(W_f \cdot [h_{(t-1)}, x_t] + b_f)$$

Εξίσωση 6: Μαθηματικός Τύπος Λειτουργίας Πύλης Λήθης

Πρακτικά συνενώνει την νέα είσοδο με την προηγούμενη κρυφή κατάσταση του επιπέδου, πολλαπλασιάζει το διάνυσμα που θα προκύψει με έναν πίνακα βαρών που αντιστοιχεί σε αυτήν την πύλη, προσθέτει το *bias* και τέλος το αποτέλεσμα περνάει από την συνάρτηση ενεργοποίησης *sigmoid*. Η *sigmoid* δημιουργεί ένα διάνυσμα που αποτελείται από τιμές στο διάστημα (0, 1), που πολλαπλασιάζεται όπως φαίνεται και από την εικόνα με το προηγούμενο διάνυσμα της μνήμης $[C_{(t-1)}]$, όπου τιμές κοντά στο 0 “αφαιρούν” τις τιμές στις αντίστοιχες θέσεις του διανύσματος μνήμης ενώ τιμές κοντά στο 1 τις διατηρούν. [7]

Πύλη Εισόδου:

Αρχικά, όπως φαίνεται και στο σχήμα, η συγκεκριμένη πύλη εκτελεί την ίδια διαδικασία με την πύλη λήθης χρησιμοποιώντας όμως τον δικό της πίνακα βαρών (W_i) και το *bias* (b_i) παράγοντας το διάνυσμα i_t . Η ίδια διαδικασία επαναλαμβάνεται άλλη μια φορά αυτήν την φορά κάνοντας χρήση του πίνακα βαρών του κελιού μνήμης (W_c) και το *bias* αυτού (b_c), ενώ χρησιμοποιείται και η συνάρτηση ενεργοποίησης *tanh* που εξάγει αριθμούς στο διάστημα (-1, 1) έναντι της *sigmoid*, και παράγεται το διάνυσμα C'_t . Τελικά η μαθηματική σχέση που ενημερώνει το διάνυσμα της μνήμης με την σημαντική πληροφορία είναι η ακόλουθη:

$$C_t = f_t \odot C_{(t-1)} + i_t \odot C'_t$$

Εξίσωση 7: Μαθηματικός Τύπος Λειτουργίας Πύλης Εισόδου

Το γινόμενο στοιχείο προς στοιχείο στα δεξιά του συμβόλου πρόσθεσης της παραπάνω σχέσης συμβολίζει τις καινούργιες υποψήφιες τιμές, για το διάνυσμα της μνήμης, αφού η κάθε μια ξεχωριστά έχει κλιμακωθεί με βάση το βαθμό που αποφασίστηκε να ενημερωθεί. [7]

Πύλη εξόδου:

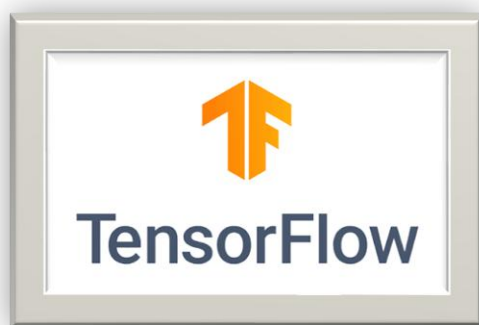
Η πύλη αυτή υπολογίζει το o_t ακολουθώντας την λογική της πύλης λήθης, χρησιμοποιώντας τον δικό της πίνακα βαρών (W_o) και το δικό της bias (b_o). Έπειτα αποφασίζεται ποια πληροφορία από το νέο κελί μνήμης (C_t) θα επιλεγεί ως η νέα κρυφή κατάσταση που θα αποτελέσει είσοδο του επιπέδου, μαζί με τα νέα δεδομένα, στο επόμενο χρονικό βήμα, από την παρακάτω μαθηματική σχέση:

$$h_t = o_t \odot \tanh(C_t)$$

Εξίσωση 8: Μαθηματικός Τύπος Λειτουργίας Πύλης Εξόδου

Τα συγκεκριμένα τεχνητά νευρωνικά δίκτυα βρίσκουν πολλές εφαρμογές, ανάμεσα τους είναι η επεξεργασία της φυσικής ανθρώπινης γλώσσας, η πρόβλεψη χρονικά συνεχόμενων τιμών π.χ. θερμοκρασία, η υλοποίηση συστημάτων εξατομικευμένων προτάσεων κ.α. [7]

2.5: TensorFlow και Keras, λίγα λόγια



Εικόνα 8: Λογότυπο βιβλιοθήκης TensorFlow

Η TensorFlow [8] είναι μια δωρεάν βιβλιοθήκη ανοιχτού κώδικα που αναπτύχθηκε από την ομάδα της Google, Google Brain, αρχικά για εσωτερική χρήση στην εταιρεία με σκοπό την έρευνα και την παραγωγή ενώ η πρώτη της έκδοση κυκλοφόρησε το 2015. Από τότε έχουν κυκλοφορήσει αρκετές βελτιωμένες επανεκδόσεις της. Η συγκεκριμένη βιβλιοθήκη χρησιμοποιείται ευρέως στον τομέα της τεχνητής νοημοσύνης και της μηχανικής μάθησης κυρίως για την εκπαίδευση και αξιολόγηση τεχνητών νευρωνικών δικτύων, διαχειρίζεται/υλοποιεί όλες τις υπολογιστικές απαιτήσεις ενός τέτοιου δικτύου. Ιδιαίτερα αξιόλογο αποτελεί ακόμα το γεγονός ότι μπορεί να χρησιμοποιηθεί με διάφορες γλώσσες προγραμματισμού όπως είναι η Python, η Java, η C++ κ.α., προσφέροντας ξεχωριστό api (Application Programming Interface) για όσες γλώσσες υποστηρίζει, βρίσκοντας εφαρμογή σε ένα ευρύ φάσμα τεχνολογικών πεδίων. Αξίζει όμως να σημειωθεί ότι η “επίσημη” γλώσσα χρήσης της

συγκεκριμένης βιβλιοθήκης είναι η *Python*, η οποία λαμβάνει και την πλήρη υποστήριξη.



Εικόνα 9: Λογότυπο Keras API

Το *Keras* [9] είναι το επίσημο, όπως προτείνει και η ίδια η ομάδα ανάπτυξης και διαχείρισης της βιβλιοθήκης *TensorFlow*, υψηλού επιπέδου, ανοιχτού κώδικα και διαπλατφορμικό (cross-platform) api για την βιβλιοθήκη *TensorFlow* γραμμένο στην γλώσσα *Python*. Επειδή σχεδιάστηκε αρχικά, πρώτος συγγραφέας ήταν ο *François Chollet*, με σκοπό να προσφέρει την δυνατότητα για γρήγορο πειραματισμό είναι ιδιαίτερα προσιτό, παραγωγικό και απλό για τον χρήστη ενώ καλύπτει όλα τα στάδια ενός προβλήματος μηχανικής μάθησης. Ακόμα προσφέρει πρόσβαση σε όλες τις δυνατότητες της βιβλιοθήκης *TensorFlow*. Παρέχει καλογραμμένα και καθαρά μηνύματα λάθους βοηθώντας τον χρήστη στην κατανόηση και διόρθωση των λαθών του ενώ παράλληλα βοηθάει γενικά στην παραγωγή ενός ακριβούς και ευκολοδιάβαστου κώδικα.

3.Εφαρμογές Μηχανικής Μάθησης για Αναγνώριση Υπνηλίας

3.1: Ταξινόμηση εφαρμογών αναγνώρισης υπνηλίας

Το πρόβλημα της αναγνώρισης της υπνηλίας κατά την οδήγηση έχει κεντρίσει το επιστημονικό ενδιαφέρον με τις εφαρμογές/συστήματα που προτείνονται ανά διαστήματα να ταξινομούνται σε τρεις βασικές κατηγορίες, σε αυτές που χρησιμοποιούν παραμέτρους της συμπεριφοράς του οδηγού, σε αυτές που χρησιμοποιούν παραμέτρους του οχήματος και σε αυτές που χρησιμοποιούν φυσιολογικές/βιολογικές παραμέτρους του οδηγού. Τα συστήματα που χρησιμοποιούν παραμέτρους του οχήματος απαιτούν την εγκατάσταση αισθητήρων σε τμήματα αυτού, όπως είναι το τιμόνι, τα πετάλια κ.α., με σκοπό να αναγνωρίσουν την κόπωση/υπνηλία του οδηγού με βάση χαρακτηριστικά του οχήματος και μοτίβα της οδήγησης όπως είναι η κλίση του τιμονιού, τα μοτίβα αλλαγής λωρίδων κ.α. Είναι περιορισμένης ακρίβειας και αποδοτικότητας συστήματα καθώς βασίζονται σε παραμέτρους οι οποίες εξαρτώνται σε σημαντικό βαθμό από τις καιρικές συνθήκες κατά την διάρκεια της οδήγησης, από την κατάσταση του δρόμου κ.α., από εξωτερικούς επί της ουσίας παράγοντες. Ακόμα ορισμένες φορές η υπνηλία του οδηγού δεν αποτυπώνεται στην κατάσταση του οχήματος και της οδήγησης, με αποτέλεσμα σε τέτοιες περιπτώσεις οι συγκεκριμένες εφαρμογές να αποτυγχάνουν, και γενικά είναι συστήματα που αντιδρούν καθυστερημένα, αφού δηλαδή ο οδηγός έχει ήδη χάσει τον έλεγχο του οχήματος. Από την άλλη τα συστήματα που βασίζονται σε φυσιολογικές/βιολογικές μετρήσεις του οδηγού, είναι πιο αξιόπιστα και αποδοτικά, τα πιο αξιόπιστα ανάμεσα στα συστήματα αναγνώρισης υπνηλίας, καθώς παρακολουθούν παραμέτρους, όπως είναι ο ρυθμός της καρδιάς, της αναπνοής κ.α., που έχουν άμεση σχέση με τον οδηγό, ειδοποιώντας τον πριν εξελιχθούν τα συμπτώματα της υπνηλίας. Βέβαια είναι σε γενικές γραμμές δύσκολο να εφαρμοστούν σε καθημερινές συνθήκες λόγω της παρεμβατικότητας τους καθώς απαιτούν ειδικό εξοπλισμό, ο οποίος θα πρέπει να είναι συνδεδεμένος με τον οδηγό, πιθανώς ενοχλώντας τον κατά την οδήγηση. Τέλος τα συστήματα που χρησιμοποιούν παραμέτρους της συμπεριφοράς του οδηγού είναι σαφώς μη παρεμβατικά εξάγοντας, μέσω παρακολούθησης του οδηγού με χρήση κάμερας και υλοποίηση τεχνικών υπολογιστικής όρασης, χαρακτηριστικά σχετικά με την συμπεριφορά του όπως είναι το χασμουρητό, το ανοιγοκλείσιμο των ματιών, η θέση του κεφαλιού, ο βαθμός κλεισίματος των ματιών κ.α. Είναι γενικά αποδοτικές εφαρμογές οι μετρήσεις όμως που χρησιμοποιούν επηρεάζονται από εξωγενείς/περιβαλλοντικούς παράγοντες όπως είναι η φωτεινότητα της καμπίνας του οχήματος, η λάμψη, το αν κοιτάει η όχι την κάμερα κ.α. [10] Το κεφάλαιο αυτό θα ασχοληθεί με εφαρμογές, αλγόριθμους, τεχνικές κ.α. της κατηγορίας ταξινόμησης, από αυτές που συζητήθηκαν παραπάνω, τα συστήματα της οποίας ασχολούνται με την παρακολούθηση παραμέτρων της συμπεριφοράς του οδηγού, καθώς και το προτεινόμενο σύστημα της παρούσας εργασίας εμπίπτει σε αυτήν την κατηγορία.

3.2: Κλασσικές μέθοδοι μηχανικής μάθησης, ορόσημα & χαρακτηριστικά

Το πρόβλημα της υπνηλίας είναι ένα πρόβλημα δυαδικής, όταν θέλουμε να ταξινομήσουμε δείγματα δεδομένων ανάμεσα σε δύο κλάσεις ξύπνιος-κοιμισμένος, ή πολυταξικής ταξινόμησης, όταν έχουμε παραπάνω από δύο κλάσεις π.χ. ξύπνιος-μισοκοιμισμένος-κοιμισμένος. Οι αλγόριθμοι κλασσικής μηχανικής μάθησης που χρησιμοποιούνται για αυτόν τον σκοπό εκπαιδεύονται και μετέπειτα αξιολογούνται και εφαρμόζονται πάνω σε χαρακτηριστικά της συμπεριφοράς του οδηγού. Ένας διαδομένος τρόπος, αλλά όχι ο μοναδικός, να υπολογιστούν αυτά τα χαρακτηριστικά είναι με χρήση ορόσημων/σημείων που εξάγονται από τα καρέ που παίρνονται από το βίντεο της κάμερας. Για χρόνια το πρότυπο για την εξαγωγή ορόσημων σε πρόσωπα ήταν η βιβλιοθήκη Dlib [11] η οποία με την χρήση ενός συνόλου δένδρων παλινδρόμησης (Ensemble of Regression Trees) εντοπίζει 68 σημεία σε αυτά σε χιλιοστά του δευτερολέπτου. Βέβαια ο αλγόριθμος που προτείνεται σε αυτήν την βιβλιοθήκη, παρότι γενικά είναι πολύ ακριβής στην πρόβλεψη των σημείων, δυσκολεύεται όταν τα πρόσωπα είναι μικρά, ή απομακρυσμένα από την κάμερα, ή το κεφάλι παρουσιάζει κλίση, ή η ποιότητα της εικόνας είναι χαμηλή. Η σύγχρονη προσέγγιση είναι πλέον η βιβλιοθήκη MediaPipe [12] από την Google. Σε αυτήν προτείνεται ένα βαθύ ελαφρύ νευρωνικό δίκτυο [13] που εντοπίζει 468 σημεία στο πρόσωπο με υψηλή ακρίβεια, προσφέροντας ένα πυκνό πλέγμα σημείων γύρω από το στόμα και τα μάτια τα οποία είναι και αυτά με την μεγαλύτερη σημασία κατά την συμπεριφορική ανάλυση και αναγνώριση της υπνηλίας, το οποίο δίκτυο είναι σχεδιασμένο να εκτελείται αποδοτικά ακόμα και σε απλές CPU και φορητές συσκευές. Αφού λοιπόν εντοπιστούν τα ορόσημα στα καρέ με βάση αυτά μπορούν να υπολογιστούν διάφορες μετρικές για την αναγνώριση της υπνηλίας. Η πιο αναγνωρισμένη και αξιόπιστη μετρική για την κατάσταση εγρήγορησης του οδηγού είναι ο δείκτης PERCLOS [14]. Επί της ουσίας ορίζεται ως το ποσοστό του χρόνου σε ένα “παράθυρο” ενός λεπτού κατά το οποίο τα μάτια του ατόμου ήταν κλειστά κατά τουλάχιστον 80%. Παρά την εγκυρότητα του ο δείκτης εμφανίζει σημαντική καθυστέρηση στην αναγνώριση της υπνηλίας καθώς απαιτεί την συσσώρευση δεδομένων αρκετών δεκάδων δευτερολέπτων πριν να μπορεί να εξάγει ένα ασφαλές και σωστό συμπέρασμα. Μια ακόμα πολύ σημαντική μετρική αποτελεί ο μαθηματικός λόγος EAR [15] ο οποίος συμβολίζει τον βαθμό ανοίγματος του ματιού σε πραγματικό χρόνο. Όταν το μάτι είναι ανοιχτό ο λόγος αυτός είναι σχετικά σταθερός σε μια τιμή ενώ όταν το μάτι είναι κλειστό είναι σχεδόν μηδενικός. Πολύ σημαντικός επίσης είναι και ο μαθηματικός λόγος MAR που υιοθετεί παρόμοια φιλοσοφία υπολογισμού με τον λόγο EAR και εκφράζει τον βαθμό ανοίγματος του στόματος σε πραγματικό χρόνο. Όταν το στόμα είναι κλειστό ο λόγος αυτός είναι σχεδόν μηδενικός ενώ η τιμή του αυξάνεται κατακόρυφα όταν το άτομο χασμουριέται. Πέραν αυτών των χαρακτηριστικών από τις συνεχείς τιμές EAR και MAR μπορούν να εξαχθούν παράγωγα στατιστικά χαρακτηριστικά όπως είναι η συχνότητα βλεφαρισμού, η συχνότητα χασμουρητού κ.α. Επόμενο και τελευταίο βασικό βήμα στην δημιουργία ενός συστήματος αναγνώρισης υπνηλίας με χρήση κλασσικών αλγορίθμων μηχανικής μάθησης και χαρακτηριστικών/μετρικών συμπεριφοράς είναι η εκπαίδευση και αξιολόγηση ενός τέτοιου αλγορίθμου. Οι μετρικές που έχουν υπολογιστεί συγκεντρώνονται σε στατικά

διανύσματα χαρακτηριστικών τα οποία αποτελούν είσοδο σε έναν εποπτευόμενο αλγόριθμο κλασσικής μηχανικής μάθησης. Όπως φαίνεται και από την έρευνα [16] ανάμεσα στους αλγορίθμους που μπορούν να χρησιμοποιηθούν, και οι οποίοι έχουν αξιόλογη επίδοση στην αναγνώριση της υπνηλίας, συγκαταλέγονται οι μηχανές διανυσμάτων υποστήριξης (Support Vector Machines - SVM) οι οποίες μάλιστα χρησιμοποιούνται ευρέως, οι k -πλησιέστεροι γείτονες (k -NN) και η λογιστική παλινδρόμηση (Logistic Regression), αλγόριθμοι που χρησιμοποιούνται κυρίως ως απλούστερα μοντέλα βάσης.

Όλοι οι κλασσικοί αλγόριθμοι μηχανικής μάθησης που μπορούν να χρησιμοποιηθούν χαρακτηρίζονται τόσο από πλεονεκτήματα όσο και από σαφή μειονεκτήματα όταν εκτελούνται σε πραγματικές συνθήκες έναντι των τεχνητών νευρωνικών δικτύων. Για αρχή λόγω της μειωμένης πολυπλοκότητας τους διακατέχονται από μικρό υπολογιστικό κόστος με αποτέλεσμα να μπορούν να εκτελεστούν και σε απλούς επεξεργαστές αλλά και σε συσκευές με γενικά περιορισμένους πόρους, κάτι το οποίο όμως δύναται να επιτευχθεί και από ένα τεχνητό νευρωνικό δίκτυο με σωστό προγραμματισμό και διαχείριση τόσο αυτού όσο και των δεδομένων. Επιπλέον προσδιορίζονται από υψηλή επεξηγησιμότητα, είναι σχετικά εύκολο να κατανοηθεί και μετέπειτα να εξηγηθεί η λειτουργία του, ενώ απαιτούν και σχετικά μικρό όγκο δεδομένων εκπαίδευσης, έναντι των τεχνητών νευρωνικών δικτύων. Από την άλλη επειδή αντιμετωπίζουν κάθε διάνυσμα χαρακτηριστικών ή χρονικό παράθυρο ως ανεξάρτητα δεδομένα αδυνατούν σε πολλές περιπτώσεις να συλλάβουν την χρονική εξάρτηση ανάμεσα στα διαφορετικά διανύσματα/παράθυρα, μην δίνοντας σημασία στην δυναμική εξέλιξη της κόπωσης/υπνηλίας στον χρόνο. Ακόμα είναι αρκετές φορές αλγόριθμοι ευαίσθητοι στον θόρυβο των δεδομένων εισόδου π.χ. κλίσεις κεφαλιού, φωτισμός κ.α. ενώ σε γενικές γραμμές επιτυγχάνουν επιδόσεις κατώτερες μοντέλων που κάνουν χρήση βαθιάς μάθησης.

Πολλά από αυτά τα μειονεκτήματα, όπως υποδηλώνει και η παραπάνω παράγραφος, καταφέρνουν να επιλύσουν άλλες φορές σε μεγαλύτερο και άλλες φορές σε μικρότερο βαθμό οι αλγόριθμοι βαθιάς μάθησης, οι οποίοι πλέον “κυριαρχούν” ανάμεσα στην επιστημονική κοινότητα για την αναγνώριση της υπνηλίας, που θα αναφερθούν στο επόμενο κεφάλαιο.

3.3: Αναφορά μεθόδων βαθιάς μάθησης

Στο πεδίο επιστημονικού ενδιαφέροντος της ανίχνευσης υπνηλίας με χρήση βαθιάς μάθησης ανάμεσα στις διαθέσιμες προσεγγίσεις, οι πιο βασικές και πιο συχνά χρησιμοποιούμενες, είναι η χρήση αποκλειστικά δισδιάστατων συνελκτικών νευρωνικών δικτύων (2D-CNNs) τα οποία δέχονται ως είσοδο καρέ του οδηγού, η χρήση αναδρομικών νευρωνικών δικτύων π.χ. LSTM αφού πρώτα έχουν υπολογιστεί συμπεριφορικά χαρακτηριστικά/μετρικές με χρήση ορόσημων ή άλλων τεχνικών και η συνδυαστική χρήση των δύο προαναφερθέντων τύπων τεχνητών νευρωνικών δικτύων, οι λεγόμενες υβριδικές προσεγγίσεις. Περνώντας σε περισσότερες λεπτομέρειες τα 2D-CNNs αποτελούν το πρότυπο στο πεδίο της υπολογιστικής όρασης για την εξαγωγή χωρικών χαρακτηριστικών από εικόνες. Στο πρόβλημα της αναγνώρισης της υπνηλίας

οι συγκεκριμένες αρχιτεκτονικές συνήθως χρησιμοποιούνται με δύο βασικούς τρόπους. Κατά τον έναν από αυτούς τα καρέ του οδηγού τροφοδοτούνται ολόκληρα σε ένα τέτοιο δίκτυο [17] και αυτό εκπαιδεύεται να εντοπίζει και να αναγνωρίζει τα, συνδυαστικά πολλές φορές, μοτίβα της υπνηλίας όπως το κλείσιμο των ματιών, το χαμουρητό κ.α., ταξινομώντας τελικώς τα καρέ σε κλάσεις. Κατά την δεύτερη προσέγγιση [18] πρώτα εντοπίζεται το πρόσωπο και απομονώνονται συγκεκριμένες περιοχές από αυτό, συνήθως τα μάτια ή/και το στόμα, και έπειτα ένα συνελκτικό νευρωνικό δίκτυο δέχεται αυτά τα αποκόμματα και τα ταξινομεί σε κλάσεις π.χ. ανοιχτό-κλειστό μάτι/στόμα. Πολύ συχνή και αποτελεσματική είναι και η χρήση LSTMs [19] επειδή η υπνηλία δεν είναι μια στατική κατάσταση αλλά ένα σταδιακά χρονικά εξελισσόμενο φαινόμενο. Κατά την αναγνώριση της υπνηλίας με την χρήση της συγκεκριμένης αρχιτεκτονικής τα χαρακτηριστικά που έχουν υπολογισθεί/εξαχθεί από τα καρέ του βίντεο της κάμερας συγκεντρώνονται σε διανύσματα τα οποία συμπληρώνουν χρονικά παράθυρα π.χ. των 20 διανυσμάτων και τροφοδοτούνται στο δίκτυο με απώτερο σκοπό αυτό να “μάθει” την εξέλιξη και χρονική εξάρτηση της υπνηλίας. Τέλος η τρίτη αρχιτεκτονική που αναφέρθηκε, η υβριδική [17], αξιοποιεί τόσο την χωρική ανάλυση που πραγματοποιούν τα CNNs στα δεδομένα όσο και την χρονική που πραγματοποιούν τα αναδρομικά νευρωνικά δίκτυα. Σε μια τυπική τέτοια αρχιτεκτονική πρώτα ακολουθίες από καρέ του βίντεο εισάγονται σε ένα 2D-CNN το οποίο εξάγει χωρικά χαρακτηριστικά από αυτά μετατρέποντας κάθε καρέ σε ένα συμπαγές διάνυσμα τιμών που κωδικοποιεί την πληροφορία που έχει εξαχθεί. Έπειτα τα διανύσματα που παράχθηκαν συγκεντρώνονται σε ένα “παράθυρο” και τροφοδοτούν ένα ή περισσότερα επίπεδα LSTM για να μοντελοποιήσουν τις χρονικές εξαρτήσεις και μεταβολές σε αυτά. Στο τέλος του δικτύου ένα τελικό πλήρως συνδεδεμένο στρώμα (Fully Connected Layer) με συνάρτηση ενεργοποίησης sigmoid ή softmax παράγει τις τελικές πιθανότητες για τα δείγματα. Συμπερασματικά οι υβριδικές προσεγγίσεις πετυχαίνουν πολύ σημαντικά ποσοστά ακρίβειας και γενικά υψηλές μετρικές με αντάλλαγμα όμως το υψηλό υπολογιστικό τους κόστος.

3.4: Σύγκριση CNNS έναντι LSTMs στην αναγνώριση της υπνηλίας

Η σύγκριση ανάμεσα σε αυτές τις δύο αρχιτεκτονικές, τις πιο βασικές και απλές στην αναγνώριση της υπνηλίας, παρουσιάζει ιδιαίτερο ενδιαφέρον. Για αρχή είναι σημαντικό να διευκρινιστεί ότι και οι δύο αρχιτεκτονικές είναι ιδιαίτερα αποδοτικές και πετυχαίνουν υψηλές μετρικές, η επιλογή ανάμεσα στις δύο είναι στην κρίση του προγραμματιστή/μηχανικού και εξαρτάται κυρίως από δευτερεύοντες παράγοντες. Για παράδειγμα λόγω της επεξεργασίας ολόκληρων ακατέργαστων καρέ έναντι της επεξεργασίας διανυσμάτων απλών τιμών τα CNNs γενικά απαιτούν υψηλή υπολογιστική ισχύ από ότι τα LSTMs. Ακόμα γενικά χρειάζονται μεγαλύτερο όγκο δεδομένων από τα LSTMs για να εκπαιδευτούν σωστά ενώ χαρακτηρίζονται και από χαμηλότερη επεξηγησιμότητα, λειτουργούν θα λέγαμε σαν “μαύρα κουτιά”. Επίσης παρουσιάζουν μεγαλύτερη ευαισθησία στον θόρυβο των δεδομένων, π.χ. αλλαγές στον φωτισμό, κλίσεις κεφαλιού κ.α., από την στιγμή που επεξεργάζονται ακατέργαστα pixel, μειονέκτημα από το οποίο σε μεγάλο βαθμό απαλλάσσονται τα LSTMs ειδικά

αν τα χαρακτηριστικά έχουν υπολογιστεί με χρήση ορόσημων. Επιπλέον επειδή τα συνελκτικά νευρωνικά δίκτυα πραγματοποιούν αποκλειστικά χωρική επεξεργασία των καρτέ, ενώ η υπνηλία είναι ένα χρονικά εξαρτώμενο φαινόμενο, μπορούν να οδηγήσουν σε λανθασμένες προβλέψεις, δεν γνωρίζουν π.χ. αν ένα κλειστό μάτι σε ένα καρτέ είναι φυσιολογικό βλεφάρισμα ή μικρο-ύπνος. Σε αντίθεση τα LSTMs αναγνωρίζουν την χρονική δυναμική των δεδομένων και παρουσιάζουν ανθεκτικότητα σε διακυμάνσεις αυτών καθώς επεξεργάζονται όχι μεμονωμένα καρτέ αλλά ένα σύνολο, χρονικό παράθυρο, από διανύσματα τιμών. Βέβαια αυτό εισάγει μια μικρή καθυστέρηση στην πρώτη πρόβλεψη σε πραγματικό περιβάλλον, αφού θα πρέπει πρώτα να συγκεντρωθεί ένα σύνολο δεδομένων. Επιπροσθέτως στα LSTMs η ανάγκη για προ επεξεργασία των δεδομένων είναι υπαρκτή και εκτενής, σε σχέση με τα CNNs που είναι σαφώς πιο περιορισμένη, καθώς θα πρέπει να εξαχθούν τα χαρακτηριστικά πάνω στα οποία θα βασίζονται οι προβλέψεις είτε με χρήση ορόσημων είτε με άλλες τεχνικές. Τέλος στα CNNs η εκτέλεση για κάθε καρτέ απαιτεί πολύ μικρό χρόνο αν υπάρχει ο κατάλληλος εξοπλισμός ώστε η διαδικασία να παραλληλιστεί ανάμεσα στις GPU.

3.5: Datasets, μετρικές αξιολόγησης και υπαρκτές προκλήσεις

3.5.1: Αναφορά στα πιο γνωστά δημόσια σύνολα δεδομένων

Για την αναγνώριση της υπνηλίας από εφαρμογές που χρησιμοποιούν τεχνικές υπολογιστικής όρασης και αξιολογούν παραμέτρους της συμπεριφοράς του οδηγού χρησιμοποιούνται κυρίως δημόσια σύνολα δεδομένων (Datasets), τόσο για την εκπαίδευση αλλά και την αξιολόγηση των μοντέλων, αλλά μπορούν να χρησιμοποιηθούν και σύνολα δεδομένων δημιουργημένα προσωπικά από τους ερευνητές του εκάστοτε συστήματος. Τα πιο διαδεδομένα δημόσια σύνολα δεδομένων στην βιβλιογραφία περιγράφονται συνοπτικά παρακάτω:

YawDD:

Το συγκεκριμένο dataset [20] χρησιμοποιείται κυρίως για την αναγνώριση του χασμουρητού και περιέχει βίντεο βιντεοσκοπημένα σε πραγματικές συνθήκες με διάφορους φωτισμούς και είναι χωρισμένο σε δύο υποσύνολα. Στο πρώτο υποσύνολο, που αποτελείται και από τα συντριπτικά περισσότερα βίντεο, η κάμερα είναι τοποθετημένη κάτω από τον μπροστινό κεντρικό καθρέπτη του αυτοκινήτου ενώ στο δεύτερο είναι τοποθετημένη πάνω στο ταμπλό του οδηγού, σε κάθε περίπτωση ο/η εθελοντής κάθεται στην θέση του οδηγού. Έχουν επιλεγεί εθελοντές και από τα δύο φύλα, διαφόρων ηλικιών και εθνικοτήτων, με διαφορετικά χαρακτηριστικά προσώπου και που φοράνε ή δεν φοράνε γυαλιά ηλίου/οράσεως, με σκοπό το σύνολο δεδομένων να εμπλουτιστεί με ποικιλομορφία. Κάθε εθελοντής έχει προσφέρει 3-4 βίντεο όπου στο καθένα βρίσκεται σε διαφορετική κατάσταση, μιλάει ή χασμουριέται ή βρίσκεται σε απάθεια.

UTA-RLDD:

Το συγκεκριμένο dataset [21] περιέχει 180 RGB βίντεο από 60 διαφορετικά υγιή άτομα με συνολική διάρκεια βιντεοσκοπήσεων περίπου 30 ώρες. Σχεδιάστηκε με στόχο κυρίως την πολυταξική ταξινόμηση της υπνηλίας καθώς δεν αποτελείται αποκλειστικά από βίντεο που είναι προφανή τα σημάδια της αλλά και από βίντεο που τα σημάδια είναι πιο μέτριας έντασης. Περιλαμβάνει βίντεο από γυναίκες αλλά και από άντρες, με τους άντρες να είναι συντριπτικά περισσότεροι, από διάφορες εθνικότητες, με τα άτομα σε κάποια από τα βίντεο να φοράνε και γυαλιά οράσεως. Κάθε εθελοντής/εθελόντρια δημιούργησε μόνος/η του/της 3 βίντεο, με το καθένα να ανήκει σε μια από τις τρεις κλάσεις εγρήγορση-χαμηλή εγρήγορση-υπνηλία. Τέλος τα βίντεο που μπορούν να βρεθούν στο συγκεκριμένο δημόσιο σύνολο δεδομένων χαρακτηρίζονται από διαφορετικές γωνίες λήψης, διαφορετικές πραγματικές συνθήκες και φόντα.

NTHU-DDD:

Το συγκεκριμένο dataset [22] περιέχει συνολικά περίπου 9.5 ώρες βίντεο από 36 διαφορετικά άτομα, χωρισμένα στην μέση σε άντρες και γυναίκες, διαφόρων εθνικοτήτων. Οι συμμετέχοντες για την δημιουργία των βίντεο, και την προσομοίωση των συνθηκών οδήγησης, κάθονταν σε κάθισμα οχήματος και έπαιζαν ένα ηλεκτρονικό παιχνίδι αγώνων ταχύτητας με χρήση προσομοιωμένου τιμονιού και πεταλιών. Κατά την διάρκεια της διαδικασίας παράχθηκαν βίντεο, τόσο με φωτισμό ημέρας αλλά και νύχτας, με τα άτομα να εκτελούν διάφορες αντιδράσεις-συμπεριφορές που τους ζητήθηκαν όπως ομιλία, γέλιο, αργό βλεφάρισμα, εγρήγορση, ύπνος κ.α. Τελικώς τα βίντεο που συγκεντρώθηκαν ταξινομούνται σε 5 σενάρια-κλάσεις, άτομο με, “γυμνό” πρόσωπο, γυαλιά οράσεως, γυαλιά ηλίου, “γυμνό” πρόσωπο σε νυχτερινό φωτισμό και γυαλιά οράσεως σε νυχτερινό φωτισμό.

3.5.2: Σημαντικότητα dataset & μετρικές αξιολόγησης

Η επιλογή του συνόλου δεδομένων τόσο για την εκπαίδευση όσο και για την αξιολόγηση του αλγορίθμου που έχει επιλεγεί για την αναγνώριση της υπνηλίας είναι ιδιαίτερα σημαντική. Το σύνολο δεδομένων πρέπει να εμπεριέχει ποιοτικά και αληθοφανή βίντεο, να διακατέχεται από ποικιλομορφία και ο όγκος των δεδομένων να είναι επαρκής αναλόγως το είδος της μεθόδου που έχει επιλεγεί για την αναγνώριση. Εκτός του συνόλου δεδομένων σημαντικό ρόλο, στην αξιολόγηση του μοντέλου, κατέχουν οι μετρικές αξιολόγησης, μαθηματικοί τύποι που “εξαρτώνται” από τις προβλέψεις αυτού και τις πραγματικές ετικέτες-κλάσεις των δειγμάτων δεδομένων και παράγουν μια τιμή. Οι τιμές αυτές εκφράζουν επί της ουσίας τις επιδόσεις του μοντέλου, βοηθώντας τον προγραμματιστή-μηχανικό να καταλάβει σε ποιους τομείς της αναγνώρισης το μοντέλο είναι ικανοποιητικό και σε ποιους πιθανόν χρειάζεται βελτίωση, π.χ. ένα μοντέλο μπορεί να αναγνωρίζει με μεγάλη επιτυχία τις καταστάσεις εγρήγορσης στους οδηγούς αλλά να δυσκολεύεται με την αναγνώριση αυτών που φανερώνουν υπνηλία.

Στο πρόβλημα της αναγνώρισης της υπνηλίας οι μετρικές οι οποίες συνήθως χρησιμοποιούνται και έχουν και την μεγαλύτερη αξία είναι οι ακόλουθες, Accuracy, Precision, Recall, F1-Score. Ακόμα χρήσιμη είναι και η δημιουργία ενός Confusion Matrix, και η μετέπειτα οπτική απεικόνιση του, ένας πίνακας ο οποίος αποτυπώνει την γενική ικανότητα του μοντέλου στο πρόβλημα της κατηγοριοποίησης. Περισσότερες πληροφορίες για τις μετρικές, τους μαθηματικούς τύπους και την σημασία τους, καθώς και για τον πίνακα που αναφέρθηκε θα δοθούν στο κεφάλαιο της πειραματικής αποτίμησης.

3.5.3: Υπαρκτές προκλήσεις στην αναγνώριση της υπνηλίας

Παρ' όλη την σημαντική βελτίωση που έχει παρατηρηθεί στα συστήματα αναγνώρισης υπνηλίας τα τελευταία χρόνια, το συγκεκριμένο πεδίο παραμένει ενεργό ερευνητικό και μηχανικό πρόβλημα καθώς κάποιες προκλήσεις είναι ακόμα υπαρκτές, μερικές από αυτές περιγράφονται στην συνέχεια. Μια πολύ μεγάλη πρόκληση αποτελεί ο έντονος νυχτερινός φωτισμός και οι απότομες αλλαγές φωτισμού καθώς έτσι αλλοιώνεται η ευκρίνεια των χαρακτηριστικών του προσώπου. Επιπλέον πρόβλημα αποτελεί η χρήση γυαλιών ηλίου με έντονη-σκούρα επίστρωση καθώς καθιστούν την παρακολούθηση των ματιών πρακτικά αδύνατη, όπως και η επικάλυψη γενικά κρίσιμων περιοχών του προσώπου με αντικείμενα όπως καπέλα, κασκόλ, μάσκες ακόμα και με τα χέρια του οδηγού. Επίσης απότομες κινήσεις του κεφαλιού π.χ. για τον έλεγχο των πλαϊνών καθρεπτών του αυτοκινήτου μπορούν να οδηγήσουν σε εσφαλμένες θετικές προβλέψεις, καθώς μπορεί να αποκρυφτούν τα μάτια ή το μοντέλο να ερμηνεύσει την κλίση του κεφαλιού ως πτώση λόγω υπνηλίας. Επιπρόσθετα τα συστήματα που γενικά υλοποιούνται αντιδρούν πρακτικά καθυστερημένα όταν η υπνηλία έχει ήδη εξελιχθεί, π.χ. τα μάτια έχουν ήδη κλείσει, χωρίς να είναι σε θέση τις περισσότερες φορές να αναγνωρίσουν την υπνηλία στα αρκετά πρώιμα στάδια της και να την αποτρέψουν από το να εξελιχθεί. Σημαντικό είναι επίσης το γεγονός ότι τα περισσότερα δημόσια σύνολα δεδομένων που προσφέρονται για την ανάπτυξη και αξιολόγηση συστημάτων αναγνώρισης υπνηλίας έχουν δημιουργηθεί σε περιβάλλοντα προσομοίωσης, με τα άτομα να παριστάνουν την υπνηλία, ενώ σε πραγματικές συνθήκες το φαινόμενο αυτό είναι πιο σύνθετο, τόσο στην εξέλιξη όσο και στην εμφάνιση του, δημιουργώντας ένα “χάσμα” ανάμεσα στην εκπαίδευση και στον πραγματικό κόσμο. Τέλος επειδή οι αλγόριθμοι που πραγματοποιούν την αναγνώριση πρέπει να εκτελούνται τοπικά σε μονάδες του αυτοκινήτου σε πραγματικό χρόνο και με σαφώς περιορισμένους πόρους η χρήση απαιτητικών υπολογιστικά συστημάτων, που πιθανόν θα έχουν και υψηλές επιδόσεις, περιορίζεται σημαντικά.

Όλες αυτές οι προκλήσεις οδηγούν σταδιακά τους ερευνητές, σε μια προσπάθεια αντιμετώπισης τους, στην ανάπτυξη πολυτροπικών συστημάτων. Οι προσεγγίσεις αυτές συνδυάζουν τα οπτικά δεδομένα από την κάμερα με τα δεδομένα συμπεριφοράς του αυτοκινήτου ή/και με τα βιολογικά/φυσιολογικά δεδομένα για την κατάσταση του οδηγού από αισθητήρες σε αυτόν.

3.6: Τοποθέτηση της παρούσας εργασίας

Με βάση την μέχρι τώρα βιβλιογραφία η εργασία που θα παρουσιαστεί λεπτομερώς στο επόμενο κεφάλαιο αποφασίστηκε να ακολουθήσει την εξής μεθοδολογία. Αρχικά το ελαφρύ νευρωνικό δίκτυο της MediaPipe εφαρμόστηκε στα καρέ των βίντεο του συνόλου δεδομένων για την εξαγωγή των γεωμετρικών σημείων του προσώπου με ακρίβεια και σε γρήγορο χρόνο. Κατόπιν οι μαθηματικοί-γεωμετρικοί δείκτες EAR και MAR υπολογίστηκαν με βάση επιλεγμένα ορόσημα του προσώπου συντελώντας ένα διάνυσμα χαρακτηριστικών 2 βαθμωτών τιμών για κάθε ξεχωριστό καρέ που επεξεργάστηκε. Το σύνολο των διανυσμάτων αυτών μοιράστηκε σε χρονικά παράθυρα σταθερού μεγέθους τα οποία αποτέλεσαν είσοδο σε ένα τεχνητό αναδρομικό νευρωνικό δίκτυο βασισμένο σε δύο στοιβαγμένα επίπεδα *Bidirectional LSTM*. Επιλέχθηκε η χρήση *LSTMs* με στόχο, από την στιγμή που η υπνηλία είναι μια χρονικά εξαρτώμενη διαδικασία, να εκμεταλλευτεί η ικανότητα τους να αναγνωρίζουν και να μαθαίνουν χρονικές εξαρτήσεις στα δεδομένα σε βάθος χρόνου σε συνδυασμό με τις χαμηλές απαιτήσεις τους σε υπολογιστικούς πόρους.

Σημείωση: Όλοι οι κώδικες της παρούσας εργασίας έχουν υλοποιηθεί στην γλώσσα προγραμματισμού Python

4. Το Προτεινόμενο Σύστημα

4.1: Φάση 1 - Δημιουργία αρχείου κειμένου από τα βίντεο του dataset

4.1.1: Περιγραφή μαθηματικών λόγων EAR & MAR

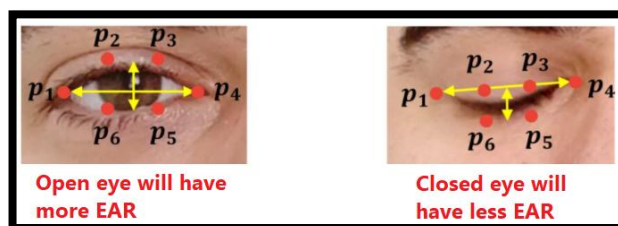
Πριν περάσουμε σε περισσότερες λεπτομέρειες για την διαδικασία δημιουργίας του αρχείου κειμένου (.csv), η οποία αποτέλεσε και το πρώτο βήμα υλοποίησης της παρούσας εργασίας, από όλα τα βίντεο που επιλέχθηκαν για το σύνολο δεδομένων, περισσότερες πληροφορίες για αυτό θα δοθούν στο κεφάλαιο της “Πειραματικής Αποτίμησης”, θα περιγραφούν αναλυτικότερα οι μαθηματικοί λόγοι EAR και MAR καθώς αποτέλεσαν τον “πυρήνα” της μετατροπής αυτής.

EAR:

Ο μαθηματικός-γεωμετρικός λόγος EAR [15] υπολογίζεται από τις ευκλείδειες αποστάσεις μεταξύ επιλεγμένων σημείων των άνω και κάτω βλεφάρων σε σχέση με την ευκλείδεια απόσταση μεταξύ των άκρων του ματιού (το πλάτος του). Γενικά η τιμή του λόγου αυτού όταν τα μάτι είναι ανοιχτό διαφέρει ελαφρώς από άτομο σε άτομο είναι δηλαδή εξαρτώμενη από τα ανατομικά χαρακτηριστικά του κάθε ανθρώπου. Ο μαθηματικός τύπος για τον υπολογισμό καθώς και οι θέσεις των σημείων του ματιού που χρησιμοποιούνται φαίνονται παρακάτω:

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Εικόνα 10: Μαθηματικός Τύπος EAR



Εικόνα 11: Θέσεις Σημείων Ματιών

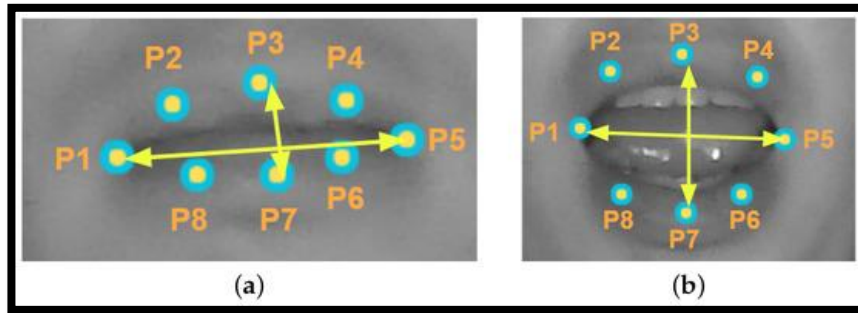
Ο υπολογισμός χρησιμοποιεί παρόμοια σημεία τόσο για το δεξί όσο και για το αριστερό μάτι, ενώ η ονοματοδοσία των σημείων σε αυτόν είναι καθαρά συμβολική. Το πως διαμορφώθηκε ο τύπος και για το δεξί αλλά και για το αριστερό μάτι, στην παρούσα εργασία με βάση τα σημεία που εντοπίζει η MediaPipe, θα αναφερθεί στην συνέχεια του κεφαλαίου 4.1.

MAR:

Ο συγκεκριμένος μαθηματικός-γεωμετρικός λόγος [19] υιοθετεί παρόμοια λογική υπολογισμού με τον λόγο *EAR*. Εξαρτάται από τις ευκλείδειες αποστάσεις μεταξύ σημείων του άνω και κάτω χείλους όπως και από την οριζόντια απόσταση μεταξύ των άκρων των χείλιων. Χρησιμοποιούνται σημεία από το εσωτερικό των χείλιων για να είναι όσο γίνεται πιο ακριβής ο υπολογισμός λόγω των ανατομικών διαφορών του στόματος ανάμεσα στα άτομα. Παρακάτω δίνεται ο τύπος υπολογισμού και οι σχετικές θέσεις των σημείων στο στόμα που εμφανίζονται σε αυτόν:

$$MAR = \frac{\|p_2 - p_8\| + \|p_3 - p_7\| + \|p_4 - p_6\|}{3 \cdot \|p_1 - p_5\|}$$

Εξίσωση 9: Μαθηματικός Τύπος MAR



Εικόνα 12: Θέσεις Σημείων Στόματος

Όπως και στον τύπο για τον *EAR* τα ονόματα των σημείων είναι καθαρά συμβολικά, το πως διαμορφώθηκε ο τύπος με βάση τα σημεία που επιστρέφει η *MediaPipe* βιβλιοθήκη θα αναφερθεί στην συνέχεια του κεφαλαίου 4.1.

4.1.2: Διαδικασία δημιουργίας αρχείου κειμένου

Η δημιουργία του αρχείου κειμένου (.csv) από το σύνολο δεδομένων, το οποίο αποτελείτε από βίντεο, πραγματοποιήθηκε ως εξής. Δημιουργήθηκε ένας βρόγχος ο οποίος στο εσωτερικό του περιέχει όλη την υπολογιστική διαδικασία. Ο βρόγχος αυτός διέτρεχε κάθε διαθέσιμο βίντεο του συνόλου δεδομένων και με την χρήση “εργαλείων” της βιβλιοθήκης *OpenCV*, μιας βιβλιοθήκης που προσφέρει “εργαλεία” υπολογιστικής όρασης για χρήση σε κώδικες γραμμένους στην γλώσσα προγραμματισμού *Python*, αποσπούσε τα καρέ κάθε βίντεο ένας προς ένα, πριν συνεχίσει με το επόμενο βίντεο. Εδώ να σημειωθεί ότι όλα τα βίντεο του dataset ήταν περίπου στα 30fps. Στην συνέχεια ακολουθούσε η επεξεργασία των καρέ η οποία όμως

πραγματοποιούνταν με ρυθμό 6fps, επεξεργαζόταν 1 καρτέ κάθε 5, με βάση την παρακολούθηση ενός μετρητή που υιοθετήθηκε για αυτόν τον σκοπό. Αποφασίστηκε ο ρυθμός επεξεργασίας να μειωθεί σε σχέση με τον ρυθμό των βίντεο, για να παραχθούν λιγότερα δεδομένα, με στόχο την μείωση του μετέπειτα υπολογιστικού κόστους του μοντέλου, τόσο κατά την εκπαίδευση αυτού όσο και κατά την εφαρμογή του. Αυτό αποφασίστηκε διότι στα περίπου 30fps που ήταν αρχικά τα βίντεο κάθε καρτέ απέχει χρονικά από το άλλο μόλις περίπου 0.03 δευτερόλεπτα, κατά συνέπεια οι διαφορές στο περιεχόμενο ανάμεσα σε διαδοχικά ή ακόμα και κοντινά χρονικά καρτέ είναι ελάχιστες. Το γιατί επιλέχθηκε τελικώς ρυθμός επεξεργασίας ίσος με 6fps και όχι κάποια διαφορετική τιμή θα εξηγηθεί στο κεφάλαιο της “Πειραματικής Αποτίμησης”. Συνεχίζοντας κάθε 1 στα 5 καρτέ λοιπόν “τροφοδοτούταν” στην βιβλιοθήκη MediaPipe της Google, αφού πρώτα είχε υποστεί κάποιες επεξεργασίες που ήταν απαραίτητες για να μπορέσει έπειτα να διαχειριστεί από αυτήν και τα τεχνητά νευρωνικά δίκτυα της. Για να μπορέσει να πραγματοποιηθεί η διαχείριση του καρτέ από την βιβλιοθήκη και η πρόβλεψη των σημείων-ορόσημων του προσώπου σε αυτό, αρχικοποιήθηκε και δημιουργήθηκε ένα έργο, task όπως το αποκαλούν και οι δημιουργοί της βιβλιοθήκης, με βάση τις απαιτήσεις της εργασίας. Συγκεκριμένα το έργο που χρησιμοποιήθηκε ονομάζεται Face Landmarker task, επί της ουσίας δημιουργήθηκε τελικώς ένα αντικείμενο της κλάσης FaceLandmarker το οποίο χρησιμοποιήθηκε για να πραγματοποιηθεί η πρόβλεψη των σημείων. Για την πρόβλεψη εκμεταλλευτήκαν τα 2 από τα 3 μοντέλα που είναι διαθέσιμα για αυτήν την διαδικασία, με το πρώτο να εντοπίζει το πρόσωπο και το δεύτερο να προβλέπει τα σημεία σε αυτό. Το τρίτο μοντέλο παίρνει ως είσοδο τα σημεία που δίνει ως έξοδο το δεύτερο και τα χρησιμοποιεί για να αναγνωρίσει χαρακτηριστικά προσώπου και εκφράσεων, κάτι το οποίο δεν χρειαζόταν στην συγκεκριμένη εργασία. Περισσότερες πληροφορίες για αυτό το task καθώς και για τον τρόπο αρχικοποίησης, δημιουργίας και χρήσης του με βάση το api της MediaPipe και τις απαιτήσεις της εκάστοτε εφαρμογής δίνονται στον οδηγό της Google [23]. Συνεχίζοντας την περιγραφή της διαδικασίας δημιουργίας του αρχείου κειμένου τα ορόσημα που προέβλεψε το μοντέλο τελικώς βρίσκονταν μέσα σε μια Python λίστα, μια δομή δεδομένων όπου τα στοιχεία είναι σειριακά αποθηκευμένα και αριθμημένα. Από αυτήν την λίστα για τον υπολογισμό των EAR/MAR χρησιμοποιήθηκαν οι δείκτες συγκεκριμένων σημείων. Ειδικότερα το πως διαμορφώθηκαν οι τύποι με βάση τους δείκτες των σημείων στην λίστα φαίνεται παρακάτω, ενώ στην θέση κάθε δείκτη στους τύπους προφανώς τοποθετήθηκε το αντίστοιχο σημείο:

$$EAR_r = \frac{dist(385, 380) + dist(387, 373)}{2 * dist(263, 362)}$$

Εξίσωση 10: Μαθηματικός Τύπος EAR για το Δεξί Μάτι με Βάση Σημεία της MediaPipe

$$EAR_l = \frac{dist(160, 144) + dist(158, 153)}{2 * dist(33, 133)}$$

Εξίσωση 11: Μαθηματικός Τύπος EAR για το Αριστερό Μάτι με Βάση Σημεία της MediaPipe

$$MAR = \frac{dist(81, 178) + dist(13, 14) + dist(311, 402)}{3 * dist(61, 291)}$$

Εξίσωση 12: Μαθηματικός Τύπος MAR με Βάση Σημεία της MediaPipe

Με το EAR με δείκτη r να συμβολίζει τον λόγο για το δεξί μάτι ενώ με δείκτη l δηλώνεται ο αντίστοιχος λόγος για το αριστερό μάτι. Η έκφραση $dist(\text{δείκτης1}, \text{δείκτης2})$ περιγράφει την ευκλείδεια απόσταση μεταξύ 2 σημείων. Εδώ να σημειωθεί ότι το μοντέλο επιστρέφει κανονικοποιημένα σημεία με τριοδιάστατες συντεταγμένες (x, y, z) για τους υπολογισμούς όμως των λόγων χρειάστηκαν μόνο οι συντεταγμένες των αξόνων x και y κάθε σημείου, ενώ επιπρόσθετα υπέστησαν αποκανονικοποίηση πριν χρησιμοποιηθούν. Τελικά ο λόγος EAR εκφράστηκε ως η μέση τιμών των 2 επιμέρους λόγων των ματιών. Στην συνέχεια δίνονται εικόνες όπου φαίνονται οι θέσεις των σημείων στα μάτια και το στόμα που προβλέπει το μοντέλο της MediaPipe και χρειάστηκαν για τους υπολογισμούς των λόγων. Οι εικόνες που οπτικοποιούν τις θέσεις των απαραίτητων σημείων για την εργασία αποτελούν ορισμένα καρτέ που με την χρήση της βιβλιοθήκης OpenCV αποσπάστηκαν από ένα επιλεγμένο βίντεο του dataset και σχεδιάστηκαν πάνω σε αυτά κύκλοι στις συντεταγμένες των οροσήμων. Ακόμα πάνω στα καρτέ έχουν σχεδιαστεί και οι αντίστοιχες τιμές των λόγων.



Εικόνα 13: Ανοιχτά Μάτια – Κλειστό Στόμα



Εικόνα 14: Κλειστά Μάτια – Κλειστό Στόμα



Εικόνα 15: Κλειστά Μάτια – Ανοιχτό Στόμα

Αφού λοιπόν ολοκληρωνόταν η διαδικασία της επεξεργασίας, με τα δεδομένα από το καρέ που επεξεργάστηκε, δημιουργούνταν μια γραμμή-εγγραφή σε ένα αρχείο κειμένου. Το αρχείο αυτό εγγραφόταν μέχρι το τέλος της επεξεργασίας όλων των

βίντεο, το τέλος δηλαδή του βρόγχου, αποκτώντας τελικώς συνολικό μέγεθος 16.911 εγγραφών, με κάθε εγγραφή να αφορά ένα καρτέ. Παρακάτω δίνεται ένα απόσπασμα από αυτό το αρχείο:

	A	B	C	D	E	F	G	H	I
1	Video Identifier,Video Name,Frame Number,Timestamp,EAR,MAR,Subject,State								
2	Video_001,1-Subject_1.avi,0,0,0,0.38190820264118275,0.011138495221762263,1,0								
3	Video_001,1-Subject_1.avi,5,0.166,0.33261472269921766,0.011746822180118546,1,0								
4	Video_001,1-Subject_1.avi,10,0.333,0.1827761580232649,0.008568528753928108,1,0								
5	Video_001,1-Subject_1.avi,15,0.5,0.29407193280135213,0.004275239443412808,1,0								
6	Video_001,1-Subject_1.avi,20,0.667,0.31836936701933005,0.005840670568555155,1,0								
7	Video_001,1-Subject_1.avi,25,0.834,0.35700567915189085,0.00810337453851953,1,0								
8	Video_001,1-Subject_1.avi,30,1.001,0.04830452383069954,0.009331523942748342,1,0								
9	Video_001,1-Subject_1.avi,35,1.167,0.29701330154605066,0.007006551305987837,1,0								
10	Video_001,1-Subject_1.avi,40,1.334,0.3798431164165983,0.006169789816299163,1,0								

Εικόνα16: Απόσπασμα .csv Αρχείου

Με την σημασία των πεδίων-στηλών να είναι η ακόλουθη:

- *Video Identifier*: Μοναδικό αναγνωριστικό που έχει δημιουργηθεί για κάθε ξεχωριστό βίντεο
- *Video Name*: Το όνομα του βίντεο όπως ήταν στο σύνολο δεδομένων
- *Frame Number*: Ο αριθμός του καρτέ που επεξεργάστηκε κάθε φορά
- *Timestamp*: Ο χρόνος εμφάνισης του καρτέ στο βίντεο σε sec
- *EAR*: Η δεκαδική τιμή του EAR για αυτό το καρτέ
- *MAR*: Η δεκαδική τιμή του MAR για αυτό το καρτέ
- *Subject*: Μοναδική τιμή-αναγνωριστικό για κάθε άτομο που έχει δοθεί σε εγγραφές αυτού, ακόμα και από διαφορετικά βίντεο που αυτό εμφανίζεται
- *State*: Παίρνει διακριτές τιμές 0 και 1, υποδηλώνοντας αν η εγγραφή αφορά δεδομένα εγρήγορσης ή υπνηλίας αντίστοιχα

4.2: Φάση 2 – Προετοιμασία δεδομένων για το TNN

4.2.1: Εισαγωγή

Τα βασικά βήματα της προετοιμασίας των δεδομένων του αρχείου κειμένου πριν την είσοδο τους στο τεχνητό νευρωνικό δίκτυο ήταν 2, ο διαχωρισμός τους σε σύνολα εκπαίδευσης, επικύρωσης και αξιολόγησης (training-validation-testing) και η δημιουργία χρονικών παραθύρων διανυσμάτων 2 τιμών από αυτά μια για το EAR και μια για το MAR, όπως και των αντίστοιχων ετικετών κάθε παραθύρου, διαδικασία

απαραίτητη για την σωστή εκπαίδευση των αναδρομικών νευρωνικών δικτύων που θα χρησιμοποιηθούν.

4.2.2: Διαχωρισμός δεδομένων σε υποσύνολα

Ο διαχωρισμός των δεδομένων σε σύνολα πραγματοποιήθηκε όπως περιγράφεται ακολούθως. Δημιουργήθηκε μια συνάρτηση η οποία δεχόταν κάθε φορά ως παραμέτρους το *Pandas dataframe* με τα δεδομένα προς διαχωρισμό και το μέγεθος του διαχωρισμού, π.χ. μέγεθος ίσο με 0.2 σημαίνει ότι το αρχικό σύνολο θα διαμεριστεί σε 2 υποσύνολα με το ένα να έχει μέγεθος 80% του αρχικού και το άλλο 20%. Η συνάρτηση αυτή υλοποιούσε έναν βρόγχο όπου με την χρήση της κλάσης *GroupShuffleSplit* της βιβλιοθήκης *Scikit-learn*, μιας ανοιχτού κώδικα βιβλιοθήκης μηχανικής μάθησης, δημιουργούσε τυχαίους διαχωρισμούς του συνόλου που περάστηκε ως παράμετρος με βάση το ορισμένο μέγεθος του διαχωρισμού και τα διαφορετικά άτομα σε αυτό, εκμεταλλεύοντας την στήλη “*Subject*” των εγγραφών. Συγκεκριμένα κάθε διαχωρισμός πραγματοποιήθηκε με τέτοιο τρόπο ώστε εγγραφές από το σύνολο δεδομένων που ανήκουν στο ίδιο άτομο, έχουν ίδια αριθμητική τιμή στην στήλη “*Subject*”, ακόμα και από διαφορετικά βίντεο, να τοποθετηθούν στο ίδιο σύνολο και όχι να “διασκορπιστούν” ανάμεσα σε σύνολα. Αυτό είναι ιδιαίτερα σημαντικό κατά την ανάπτυξη τεχνητών νευρωνικών δικτύων καθώς για να μετρηθεί αξιόπιστα η απόδοση του μοντέλου πρέπει τα δεδομένα αξιολόγησης που θα τροφοδοτηθούν σε αυτό να είναι εντελώς νέα και άγνωστα για αυτό, αλλιώς οι μετρήσεις θα είναι πλασματικά υψηλότερες λόγω απομνημόνευσης (*overfitting*). Η εκτέλεση, συνεχίζοντας, του βρόγχου σταματούσε όταν βρισκόταν ο πρώτος διαχωρισμός που ικανοποιούσε 2 συγκεκριμένα κριτήρια. Περνώντας σε λεπτομέρειες το πρώτο από αυτό αφορούσε το πραγματικό μέγεθος, και όχι αυτό με βάση το πλήθος των διαφορετικών ατόμων που παράγει η *GroupShuffleSplit*, του συνόλου που διαχωρίστηκε από αυτό που περάστηκε ως παράμετρος, το μέγεθος του δηλαδή με βάση το πλήθος των γραμμών-εγγραφών του σε σχέση με το πλήθος των γραμμών-εγγραφών του αρχικού συνόλου, ο δεκαδικός αριθμός επί της ουσίας που προκύπτει από την πράξη πλήθος εγγραφών “αποκομμένου” συνόλου / πλήθος εγγραφών αρχικού συνόλου. Αυτός ο αριθμός έπρεπε να απέχει το πολύ +/- 0.01 από το μέγεθος του διαχωρισμού που περάστηκε ως παράμετρος στην συνάρτηση και το χρησιμοποίησε η *GroupShuffleSplit* για την δημιουργία των υποσυνόλων. Το πραγματικό μέγεθος ελέγχθηκε καθώς η *GroupShuffleSplit*, όπως αναφέρθηκε, πραγματοποιεί την διαμέριση με βάση τα διαφορετικά άτομα στο αρχικό σύνολο, το οποίο ενέχει “κινδύνους”. Έστω ότι επιλέχθηκε μέγεθος διαχωρισμού ίσο με 0.2, το “αποκομμένο” σύνολο θα περιέχει τελικώς απλά όλα τα δεδομένα από το τυχαίο 20% του πλήθους των αρχικών διαφορετικών ατόμων, με αποτέλεσμα το πραγματικό πλήθος των γραμμών να είναι πιθανό να είναι σημαντικό μικρότερο ή μεγαλύτερο από 20% σε σχέση με το πλήθος των γραμμών του αρχικού συνόλου, αναλόγως με το αν τα άτομα που επιλέχθηκαν είχαν πολλές ή λίγες εγγραφές στο αρχικό σύνολο. Η τιμή +/- 0.01 επιλέχθηκε ως περιθώριο απόκλισης του πραγματικού μεγέθους στον έλεγχο για το κατάλληλο σύνολο με σκοπό να δοθεί μεγαλύτερη ευελιξία στην επιλογή του χωρίς να “θυσιαστεί” σημαντική ακρίβεια. Ο δεύτερος έλεγχος που έπρεπε να ικανοποιηθεί για να επιλεγεί

μια διαμέριση αφορούσε το ποσοστό της υπνηλίας στο σύνολο των δεδομένων του “αποκομμένου” συνόλου. Ειδικότερα υπολογίστηκε το ποσοστό των εγγραφών, ως δεκαδικός αριθμός, με τιμή 1 στο πεδίο “State” στο σύνολο που περάστηκε ως παράμετρος στην συνάρτηση όπως και στο διαχωρισμένο από αυτό σύνολο. Τελικώς θα έπρεπε το ποσοστό στο “αποκομμένο” σύνολο να αποκλίνει κατά μέγιστο ± 0.02 από αυτό του αρχικού. Αυτό ελέγχθηκε για να διατηρηθεί η ίδια αναλογία ανάμεσα στα δεδομένα, κατά συνέπεια ανάμεσα στις 2 κλάσεις εγρήγορση-υπνηλία, μεταξύ των συνόλων που προέκυψαν, εγγυώντας κατάλληλα και ισορροπημένα σύνολα τόσο για εκπαίδευση όσο και για αξιολόγηση. Η απόκλιση της τάξης του ± 0.02 στο ποσοστό επιλέχθηκε για τον ίδιο λόγο που προτιμήθηκε και κατά τον έλεγχο του μεγέθους. Τέλος να σημειωθεί ότι η συνάρτηση αυτή κλήθηκε-εκτελέστηκε 2 φορές, την πρώτη δέχτηκε ως παράμετρο το αρχικό dataset σε μορφή dataframe και δημιούργησε ένα σύνολο εκπαίδευσης και ένα αξιολόγησης, ενώ την δεύτερη περάστηκε ως παράμετρος το σύνολο εκπαίδευσης που κατασκευάστηκε από την πρώτη κλήση και παράχθηκε ένα νέο τελικό σύνολο εκπαίδευσης καθώς και ένα επικύρωσης.

4.2.3: Δημιουργία χρονικών παραθύρων & ετικετών για το TNN

Με σκοπό την δημιουργία των χρονικών παραθύρων σταθερού μεγέθους διανυσμάτων 2 διακριτών τιμών EAR/MAR και των αντίστοιχων ετικετών τους υλοποιήθηκε μια συνάρτηση. Η συνάρτηση αυτή κλήθηκε 3 φορές μια για κάθε σύνολο “ωμών” δεδομένων που δημιουργήθηκε από την συνάρτηση που περιγράφηκε στο κεφάλαιο 4.2.2, με το εκάστοτε σύνολο να αποτελεί παράμετρο της, για να μετατραπεί το κάθε ένα σε σύνολο χρονικών παραθύρων. Η δημιουργία των παραθύρων και των ετικετών έγινε με τα δεδομένα κάθε διαφορετικού βίντεο καθενός συνόλου ξεχωριστά, εκμεταλλευόμενοι την στήλη “Video Identifier” και την μέθοδο groupBy() της βιβλιοθήκης pandas, ώστε κάθε χρονικό παράθυρο να περιλαμβάνει τιμές EAR και MAR αποκλειστικά από ένα βίντεο και ταυτόχρονα να δοθεί η σωστή ετικέτα στο κάθε ένα. Εδώ να σημειωθεί ότι για κάθε χρονικό παράθυρο εκτός από την ετικέτα του 0/1 αποθηκεύτηκε και η τιμή του πεδίου “Video Identifier” που του αναλογούσε καθώς η πληροφορία σχετικά με το σε ποιο βίντεο ανήκει το κάθε ένα παράθυρο χρειάστηκε από άλλη συνάρτηση επεξεργασίας που χρησιμοποιήθηκε σε μετέπειτα φάση του συστήματος που προτείνεται σε αυτήν την εργασία. Περνώντας τώρα σε περισσότερες λεπτομέρειες σχετικά με την διαδικασία δημιουργίας των παραθύρων πριν αυτά δημιουργηθούν εφαρμόστηκε Min-Max scaling σε όλες τις τιμές των EAR και MAR καθενός βίντεο. Αυτό έγινε για να εξαιρεθούν οι ανατομικές διαφορές ανάμεσα στα άτομα του dataset, π.χ. έστω ότι ένα άτομο με “φυσιολογικά” μάτια έχει μέγιστο EAR 0.3, ένα άτομο με φυσικά σχιστά μάτια θα έχει σαφώς χαμηλότερο μέγιστο EAR, οδηγώντας το μοντέλο να νομίζει εσφαλμένα ότι το άτομο με τα σχιστά μάτια έχει αρχίσει να εμφανίζει συμπτώματα υπνηλίας παρότι αυτό μπορεί να είναι πλήρως ξύπνιο. Η εξάλειψη όλων αυτών των ανατομικών διαφορών επιτυγχάνεται διότι με την εφαρμογή της κανονικοποίησης κάθε τιμή τόσο του λόγου EAR όσο και του MAR εκφράζεται ως προς το ωφέλιμο εύρος κίνησης των ματιών και του στόματος αντίστοιχα. Με απλά λόγια κάθε τιμή αυτών των λόγων πλέον δηλώνει επί της ουσίας το ποσοστό κατά το οποίο είναι ανοιχτά τα μάτια ή το στόμα του κάθε ατόμου, π.χ. μια

τιμή EAR ίση με 0.2 μετά από την κανονικοποίηση υποδηλώνει ότι τα μάτια αυτού του ατόμου είναι 20% ανοιχτά. Ο μαθηματικός γενικός τύπος της συγκεκριμένης κανονικοποίησης δίνεται παρακάτω:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Εικόνα 17: Τύπος Κανονικοποίησης

Οι τύποι όμως που τελικώς χρησιμοποιήθηκαν για την κανονικοποίηση των τιμών των λόγων EAR και MAR τροποποιήθηκαν ελαφρώς και δίνονται παρακάτω:

$$EAR_{scaled} = \frac{EAR - EAR_{min}}{\max(EAR_{max} - EAR_{min}, MIN_EAR_RANGE)}$$

Εξίσωση 13: Τύπος Κανονικοποίησης Λόγου EAR

$$MAR_{scaled} = \frac{MAR - MAR_{min}}{\max(MAR_{max} - MAR_{min}, MIN_MAR_RANGE)}$$

Εξίσωση 14: Τύπος Κανονικοποίησης Λόγου MAR

Με τις μέγιστες και ελάχιστες τιμές και των 2 λόγων να εντοπίστηκαν για κάθε βίντεο ξεχωριστά. Οι σταθερές τώρα MIN_EAR_RANGE και MIN_MAR_RANGE χρησιμοποιήθηκαν ως ασφαλή όρια ώστε ο παρονομαστής να μην μειωθεί υπερβολικά πολύ και πλησιάσει το μηδέν. Αν δηλαδή για κάποιο βίντεο η διαφορά της μέγιστης από την ελάχιστη τιμή κάποιου λόγου ήταν κάτω από την αντίστοιχη σταθερά επιλεγόταν ως παρονομαστής το ασφαλές όριο. Για παράδειγμα, υπήρχαν βίντεο στο dataset στα οποία καθόλη την διάρκεια τους το άτομο δεν άνοιξε καθόλου το στόμα του, αυτό είχε ως αποτέλεσμα η διαφορά στον παρονομαστή του τύπου για την κανονικοποίηση του MAR να είναι σχεδόν μηδενική δημιουργώντας πλασματικά υψηλές κανονικοποιημένες τιμές για αυτόν τον λόγο για αυτά τα βίντεο ενώ το άτομο μπορεί να ήταν πλήρως ξύπνιο, οδηγώντας σε προβληματική εκπαίδευση του μοντέλου. Σε ότι τώρα αφορά τις τιμές που επιλέχθηκαν ως ασφαλή όρια και για τους 2 λόγους αυτό έγινε με βάση κάποια στατιστικά που υπολογίστηκαν στο αρχικό συνολικό σύνολο δεδομένων. Ειδικότερα υπολογίστηκε η διαφορά της μέγιστης τιμής από την ελάχιστη για κάθε έναν από τους 2 όρους για κάθε βίντεο ξεχωριστά, με απλά λόγια υπολογίστηκε το ωφέλιμο εύρος κίνησης των ματιών και του στόματος του ατόμου σε κάθε βίντεο. Αφού υπολογίστηκαν όλες αυτές οι διαφορές βρέθηκε το μέσο

(median) και το δέκατο εκατοστημόριο ανάμεσα στα εύρη των ματιών αλλά και του στόματος ξεχωριστά. Το δέκατο εκατοστημόριο είναι μια στατιστική τιμή με την ιδιότητα πως το 90% των ευρών κίνησης, των ματιών ή του στόματος αναλόγως το σε ποιο όργανο αναφέρεται, βρίσκεται πάνω από αυτήν. Τελικώς για την κανονικοποίηση των τιμών του λόγου *EAR* χρησιμοποιήθηκε ως ασφαλές όριο μια τιμή πολύ κοντά στο δέκατο εκατοστημόριο, συγκεκριμένα επιλέχθηκε το 0.2 με αυτό να είναι 0.19, ενώ για την κανονικοποίηση των τιμών του λόγου *MAR* επιλέχθηκε μια τιμή κοντά στο μέσο, συγκεκριμένα χρησιμοποιήθηκε το 0.25 με το μέσο να είναι 0.274. Για το *MAR* δεν χρησιμοποιήθηκε το δέκατο εκατοστημόριο όπως έγινε με το *EAR* καθώς αυτό παρουσίασε πολύ χαμηλή τιμή, συγκεκριμένα 0.013, ενισχύοντας την παρατήρηση πως σε έναν όχι ασήμαντο αριθμό βίντεο το εύρος κίνησης του στόματος του ατόμου ήταν πολύ χαμηλό, σχεδόν μηδενικό. Επομένως προτιμήθηκε μια τιμή κοντά στο μέσο καθώς η επιλογή του δέκατου εκατοστημορίου δεν θα εμφάνιζε κανένα ουσιαστικό όφελος κατά την κανονικοποίηση. Συνεχίζοντας τώρα την περιγραφή της συνάρτησης δημιουργίας των παραθύρων, αφού ολοκληρώθηκε η κανονικοποίηση, χρησιμοποιήθηκε η τεχνική του “Κυλιόμενου Παραθύρου” με βάση την οποία κατασκευάστηκαν χρονικά παράθυρα σταθερού μεγέθους αποτελούμενα από διανύσματα 2 κανονικοποιημένων τιμών *EAR/MAR*. Τα παράθυρα δημιουργήθηκαν με βάση ένα σταθερό ορισμένο βήμα (stride), “πάνω” σε έναν διοδιάστατο πίνακα που περιείχε τις κανονικοποιημένες τιμές των 2 λόγων του ατόμου του τρέχοντος βίντεο, διαστάσεων (πλήθος καρέ, 2), ενώ λήφθηκε υπόψιν και η περίπτωση το τελευταίο χρονικό παράθυρο να μην ευθυγραμμιστεί με το τέλος του πίνακα, δηλαδή να “περισσέψουν” διανύσματα. Σε αυτήν την περίπτωση, για να μην “πάνε χαμένα” αυτά τα διανύσματα, χρησιμοποιήθηκε η τεχνική της “Ευθυγράμμισης Τελευταίου Παραθύρου”, επί της ουσίας παράχθηκε ένα τελευταίο παράθυρο το οποίο “ξεκινούσε” ακριβώς τόσα καρέ από το τέλος του πίνακα όσα και το μέγεθος των παραθύρων. Τέλος για την δημιουργία των ετικετών των παραθύρων, οι οποίες είναι απαραίτητες για την εκπαίδευση του μοντέλου, ακολουθήθηκε μια απλή προσέγγιση με την ετικέτα του τελευταίου καρέ κάθε χρονικού παραθύρου που φτιάχτηκε να ορίζεται ως η καθολική ετικέτα αυτού.

4.3: Φάση 3 – Αρχιτεκτονική TNN και εκπαίδευση

4.3.1: Παρουσίαση αρχιτεκτονικής προτεινόμενου μοντέλου

Για τους σκοπούς της παρούσας εργασίας και του θέματος σχεδιάστηκε μια custom αρχιτεκτονική τεχνητού νευρωνικού δικτύου, με την χρήση του Keras api που έχει αναφερθεί σε προηγούμενο κεφάλαιο. Παρακάτω ακολουθούν εικόνες όπου φαίνεται η αρχιτεκτονική αυτή καθώς και ο κώδικας δημιουργίας της:

Model: "sequential"

Layer (type)	Output Shape	Param #
bidirectional (Bidirectional)	(None, 30, 32)	2,432
layer_normalization (LayerNormalization)	(None, 30, 32)	64
dropout (Dropout)	(None, 30, 32)	0
bidirectional_1 (Bidirectional)	(None, 16)	2,624
layer_normalization_1 (LayerNormalization)	(None, 16)	32
dropout_1 (Dropout)	(None, 16)	0
dense (Dense)	(None, 16)	272
dense_1 (Dense)	(None, 1)	17

Total params: 5,441 (21.25 KB)
Trainable params: 5,441 (21.25 KB)
Non-trainable params: 0 (0.00 B)

Model's training in progress...

Εικόνα 18: Αρχιτεκτονική Προτεινόμενου Μοντέλου

```

53 # model architecture
54 my_model = Sequential()
55 my_model.add(Input(shape = (windowSize, 2)))
56 my_model.add(Bidirectional(layer = LSTM(units = 16, return_sequences = True)))
57 my_model.add(LayerNormalization())
58 my_model.add(Dropout(rate = 0.05))
59 my_model.add(Bidirectional(layer = LSTM(units = 8)))
60 my_model.add(LayerNormalization())
61 my_model.add(Dropout(rate = 0.05))
62 my_model.add(Dense(units = 16, activation = "leaky_relu"))
63 my_model.add(Dense(units = 1, activation = "sigmoid"))

```

Εικόνα 19: Κώδικας Αρχιτεκτονικής με Χρήση του Keras Api

Όπως φαίνεται και από την εικόνα το μοντέλο αποτελείται από μόλις 5.441 παραμέτρους, επιβεβαιώνοντας τον ισχυρισμό πως είναι ιδιαίτερα αποδοτικό σε ότι αφορά τις απαιτήσεις πόρων και υπολογιστικής ισχύος. Περνώντας τώρα σε τεχνικές λεπτομέρειες σε ότι αφορά την αρχιτεκτονική που επιλέχθηκε προκύπτουν τα ακόλουθα. Αρχικά επιλέχθηκαν 2 επίπεδα *Bidirectional LSTMs*, τα οποία είναι μια εξέλιξη των κλασικών *LSTMs*, που έχουν την ιδιότητα πως επεξεργάζονται τα δεδομένα των ακολουθιών-παραθύρων τόσο από την αρχή προς το τέλος όσο και από την αντίθετη φορά δίνοντας τους την δυνατότητα να “καταγράφουν” πληροφορία και από το παρελθόν και από το μέλλον. Ακριβώς αυτή η διαφοροποίηση τους από τα

κλασσικά *LSTMs* τα καθιστά ικανότερα από αυτά στην κατανόηση ακολουθιών και κατά συνέπεια στην αναγνώριση της υπνηλίας. Τεχνικά κάθε τέτοιο στρώμα αποτελείται από δύο ξεχωριστά επίπεδα *LSTM* με το ένα να αναλαμβάνει την εμπρόσθια επεξεργασία των δεδομένων των παραθύρων και το άλλο την οπίσθια. Ενώ η τελική έξοδος (η κρυφή κατάσταση) του *Bidirectional LSTM* επιπέδου σε κάθε χρονικό βήμα εντός του παραθύρου είναι η συνένωση των εξόδων των 2 επιπέδων. Όπως φαίνεται τώρα και από την εικόνα με τον κώδικα της αρχιτεκτονικής το πρώτο επίπεδο *Bidirectional LSTM* ορίστηκε με 16 τεχνητούς νευρώνες, δηλαδή κάθε ένα από τα 2 *LSTM* που το αποτελούν διαθέτει 16 νευρώνες. Αυτό έχει σαν αποτέλεσμα η έξοδος του επιπέδου σε κάθε χρονικό βήμα να είναι ένα διάνυσμα των 32 τιμών, 16 τιμές από το εμπρόσθιο *LSTM* και άλλες τόσες από το οπίσθιο, ενώ με τον ορισμό της παραμέτρου *return_sequences* σε *True* το επίπεδο αυτό θα διοχετεύσει κάθε τέτοιο διάνυσμα στο επόμενο επίπεδο επεξεργασίας όπως βλέπουμε και από την διάσταση της εξόδου που είναι ίση με (*None*, 30, 32), με το *None* να συμβολίζει το πλήθος των παραθύρων που επεξεργάστηκε ταυτόχρονα το επίπεδο (το μέγεθος του *batch* εκπαίδευσης). Το δεύτερο *Bidirectional LSTM* στρώμα ρυθμίστηκε με 8 νευρώνες παράγοντας στην έξοδο του διανύσματα των 16 τιμών διοχέτευσε όμως στο επόμενο επίπεδο του δικτύου μόνο την έξοδο του για το τελευταίο χρονικό βήμα των παραθύρων όπως υποδηλώνει και το μέγεθος της εξόδου του που είναι (*None*, 30, 16). Η διαφοροποίηση αυτή ανάμεσα στις διαστάσεις των εξόδων των 2 επιπέδων επιλέχθηκε καθώς η έξοδος του πρώτου επιπέδου αποτελεί είσοδο στο δεύτερο και τα *LSTM* γενικά “απαιτούν” η είσοδος τους να είναι τριοδιάστατη, ενώ ο αριθμός των νευρώνων διατηρήθηκε χαμηλός και στα 2 επίπεδα καθώς τα παράθυρα των δεδομένων εισόδου αποτελούνταν από διανύσματα μόλις 2 διακριτών τιμών. Επομένως κρατώντας τον αριθμό των νευρώνων χαμηλό ελαττώθηκε η πιθανότητα υπερ-εκπαίδευσης (*overfitting*) και παράλληλα το δίκτυο δημιουργήθηκε “ελαφρύ”. Συνεχίζοντας τώρα την περιγραφή της αρχιτεκτονικής παρατηρούμε πως έχουν συμπεριληφθεί και δύο επίπεδα *dropout(rate)* έπειτα από τα επίπεδα επεξεργασίας *Bidirectional LSTM* (συνηθίζεται τα επίπεδα *dropout* να τοποθετούνται έπειτα από επίπεδα επεξεργασίας). Τα συγκεκριμένα επίπεδα υλοποιούν μια διαδικασία κατά την οποία “απενεργοποιούν” τυχαία ένα σύνολο τιμών της εισόδου που δέχονται με κάθε τιμή να μηδενίζεται με βάση μια πιθανότητα ίση με *rate*. Συγκεκριμένα σε αυτήν την αρχιτεκτονική το πρώτο τέτοιο επίπεδο θα επιλέξει τυχαία τιμές για μηδενισμό, με κάθε τιμή να έχει πιθανότητα να μηδενιστεί 5%, σε κάθε διάνυσμα, από τα 32, και σε κάθε παράθυρο, από τα *None*, ξεχωριστά. Παρόμοια είναι η λειτουργία της διαδικασίας και για το δεύτερο *dropout*. Σε ότι αφορά την χρήση αυτής της μεθόδου επιλέχθηκε καθώς συνεισφέρει σημαντικά στην αποφυγή της υπερ-προσαρμογής του δικτύου βοηθώντας το να γενικεύει καλύτερα, καθώς δεν μπορεί να βασίζεται σε μεγάλο βαθμό στην έξοδο συγκεκριμένων νευρώνων ούτε επιτρέπει την “συνεργασία” αυτών, και να κατασκευάζει πιο ουσιαστικές αναπαραστάσεις των δεδομένων εισόδου. Η πιθανότητα τώρα μηδενισμού μια τιμής διατηρήθηκε πολύ χαμηλή, στο 5%, για να μην “χαθεί” πολύ πληροφορία, καθώς το δίκτυο αποτελείται από λίγους νευρώνες και επίσης τα αναδρομικά νευρωνικά δίκτυα σε γενικές γραμμές είναι “ευαίσθητα” στον έντονο μηδενισμό τιμών που παράγουν. Προχωρώντας την ανάλυση της αρχιτεκτονικής χρησιμοποιήθηκε και το επίπεδο *LayerNormalization()* αμέσως μετά από κάθε ένα από τα επίπεδα *Bidirectional LSTM* και πριν τα *dropout*. Το συγκεκριμένο επίπεδο κανονικοποιεί τις τιμές κάθε διανύσματος της εισόδου που δέχεται ξεχωριστά με βάση την μέση τιμή και την διασπορά των τιμών αυτού. Ο στόχος είναι μετά την

κανονικοποίηση η μέση τιμή καθενός διανύσματος να είναι μηδέν και η διασπορά ένα. Γενικά είναι μια πολύ ωφέλιμη τεχνική καθώς βοηθάει στην αποφυγή εμφάνισης προβλημάτων όπως τα *exploding & vanishing gradients*, διασφαλίζοντας πως όλες οι τιμές της εξόδου των επιπέδων στις οποίες εφαρμόστηκε ανήκουν αποκλειστικά σε ένα σταθερό εύρος, καθιστώντας παράλληλα την εκπαίδευση του δικτύου ομαλότερη και πιο αποτελεσματική. Ακόμα επιταχύνει την διαδικασία της εκπαίδευσης και αποτελεί σημαντική προσθήκη ειδικά για τα αναδρομικά νευρωνικά δίκτυα που επεξεργάζονται ακολουθίες. Στο τέλος τώρα του τεχνητού νευρωνικού δικτύου της εργασίας υπάρχουν 2 πλήρως συνδεδεμένα επίπεδα (*Dense*) τα οποία πραγματοποιούν και την κατηγοριοποίηση των *None* διανυσμάτων που παράγει το προηγούμενο *Bidirectional LSTM* στρώμα. Συγκεκριμένα το τελευταίο *Dense* επίπεδο με την συνάρτηση ενεργοποίησης *sigmoid* παράγει *None* τιμές στο διάστημα $(0, 1)$ με την κάθε μια να εκφράζει την πιθανότητα το αντίστοιχο παράθυρο να περιγράφει υπνηλία, να ανήκει δηλαδή στην κλάση 1, με λίγα λόγια πραγματοποιεί δυαδική ταξινόμηση. Ενώ το προηγούμενο *Dense* στρώμα εισάγει κυρίως μη γραμμικότητα στο δίκτυο εντάσσοντας στην λειτουργία του την συνάρτηση ενεργοποίησης *Leaky Relu*. Η συγκεκριμένη συνάρτηση έναντι, της κλασσικής και ευρέως χρησιμοποιούμενης στην βιβλιογραφία, *Relu* διαφέρει από αυτήν πρακτικά στο γεγονός πως αντί να μηδενίζει τελείως τις αρνητικές τιμές που δέχεται ως είσοδο τις πολλαπλασιάζει με την σταθερά 0.01 μετατρέποντας τις σε μικρούς αριθμούς. Γενικά έχει προταθεί ως βελτίωση της *Relu* καθώς επιλύει πιθανά θέματα αυτής όπως το πρόβλημα των “Νεκρών Νευρώνων”, νευρώνες που παράγουν μονίμως αρνητικές τιμές οι οποίες μηδενίζονται από την *Relu* σταματώντας την εκπαίδευση τους, ωθώντας το δίκτυο αρκετές φορές σε μια πιο γρήγορη και σταθερή εκπαίδευση, κάτι το οποίο παρατηρήθηκε και στην συγκεκριμένη αρχιτεκτονική καθώς δοκιμάστηκαν και οι 2 συναρτήσεις.

4.3.2: Παρουσίαση λεπτομερειών εκπαίδευσης

Περνώντας στις λεπτομέρειες της εκπαίδευσης ως αλγόριθμος βελτιστοποίησης χρησιμοποιήθηκε ο *Adam*, ο πιο διαδεδομένος πλέον στην βιβλιογραφία λόγω της ταχύτητας και της σταθερότητας που προσφέρει στην εκπαίδευση τεχνητών νευρωνικών δικτύων. Εδώ να σημειωθεί ότι δοκιμάστηκαν ακόμα μερικοί αλγόριθμοι όπως ο *RMSProp* και ο *AdamW*, μια παραλλαγή του κλασσικού *Adam*, χωρίς όμως κανένας από αυτούς που δοκιμάστηκαν να προσέφερε κάποιο επιπλέον ορατό όφελος στην εκπαίδευση έναντι του κλασσικού *Adam* επομένως τελικώς προτιμήθηκε αυτός. Σε ότι αφορά την συνάρτηση απώλειας προτιμήθηκε μια βελτιωμένη παραλλαγή της κλασσικής *Binary Cross Entropy*, που επιλέγεται συνήθως σε προβλήματα δυαδικής ταξινόμησης, η *Binary Focal Cross Entropy*. Το βασικό μειονέκτημα της *Binary Cross Entropy* είναι πως “ζυγίζει” τα λάθη και των 2 κλάσεων ακριβώς με τον ίδιο τρόπο, δηλαδή το σφάλμα που υπολογίζεται με βάση τον τύπο της είτε η πρόβλεψη ανήκει πραγματικά στην κλάση 0 είτε στην κλάση 1 έχει ακριβώς την ίδια βαρύτητα. Σε μη ισορροπημένα σύνολα δεδομένων, δηλαδή η μια κλάση έχει πολλαπλάσια δείγματα και συνήθως εύκολα ταξινομήσιμα έναντι της άλλης, όπως είναι ένα σύνολο δεδομένων για την αναγνώριση της υπνηλίας όπου συνήθως τα δεδομένα της υπνηλίας είναι σαφώς λιγότερα από αυτά της εγρήγορσης, όπως συνέβαινε και με το *dataset* της

παρούσας εργασίας, η *Binary Cross Entropy* δεν αποδίδει ιδανικά οδηγώντας το μοντέλο να δυσκολεύεται εμφανώς με την ταξινόμηση των δειγμάτων της μειονοτικής κλάσης, κάτι το οποίο παρουσιάστηκε αρχικά και στο προτεινόμενο μοντέλο. Αυτά τα προβλήματα λοιπόν επιλύει η *Binary Focal Cross Entropy Loss* ο τύπος της οποίας δίνεται στην συνέχεια:

$$Focal(p, y) = - \underbrace{\alpha_t}_{\substack{\text{class} \\ \text{inverse} \\ \text{weighing}}} \cdot \underbrace{(1 - p_t)^\gamma}_{\substack{\text{Probability} \\ \text{downweighing}}} \cdot \log(p_t)$$

Εικόνα 20: Μαθηματικός Τύπος της BFCE

Με το p_t να συμβολίζει την προβλεπόμενη πιθανότητα και το y την πραγματική κλάση. Η συνάρτηση αυτή εισάγει 2 παράγοντες, τον παράγοντα υποβάθμισης της πιθανότητας, στο κόκκινο πλαίσιο, και τον παράγοντα αντίστροφης στάθμισης κλάσης, στο μπλε πλαίσιο. Ο πρώτος παράγοντας, ο οποίος εμπεριέχει και το γ που είναι μια σταθερά που ορίζεται από τον προγραμματιστή (υπερ-παράμετρος), πρακτικά μειώνει την βαρύτητα του σφάλματος που υπολογίζεται για τις εύκολες-σίγουρες προβλέψεις του μοντέλου, προβλέψεις δηλαδή με πολύ υψηλή πιθανότητα, “ωθώντας” το να “εστιάσει” περισσότερο σε δύσκολα δείγματα και σε προβλέψεις που είχαν χαμηλή τιμή. Ακόμα όσο πιο μεγάλη είναι η τιμή του γ τόσο πιο πολύ μειώνεται η βαρύτητα των υψηλών πιθανοτήτων ενώ αυξάνεται αυτή των χαμηλών. Από την άλλη ο δεύτερος παράγοντας μπορεί είτε να οριστεί χειροκίνητα, με μια δεκαδική τιμή, είτε να υπολογιστεί αυτόματα με βάση το αντίστροφο της συχνότητας εμφάνισης της κάθε κλάσης στα δεδομένα. Για παράδειγμα αν η πλειοψηφική κλάση έχει συχνότητα εμφάνισης 0.8 (8/10) για τις προβλέψεις που ανήκουν σε αυτήν την κλάση ο παράγοντας αυτός θα πάρει την τιμή 0.2 ενώ για τις προβλέψεις που ανήκουν στην μειονοτική κλάση θα εκλάβει την τιμή 0.8. Αυτό δίνει ακόμα μεγαλύτερη βαρύτητα στο σφάλμα που παράγεται για τις προβλέψεις της μειονοτικής κλάσης συγκριτικά με αυτό που παράγεται για τις προβλέψεις της κυρίαρχης κλάσης, “αναγκάζοντας” το μοντέλο να μαθαίνει σταδιακά μοτίβα σχετικά με τα παραδείγματα της μειονοτικής κλάσης ενισχύοντας εντέλει την απόδοσή του.

Τώρα κατά την διάρκεια της εκπαίδευσης, με σκοπό της βελτίωσης αυτής υιοθετήθηκαν μερικές ακόμα τεχνικές. Ειδικότερα χρησιμοποιήθηκε η κλάση *EarlyStopping()* από το *Keras* api η οποία σταματάει την εκπαίδευση του μοντέλου αν ολοκληρωθεί ένας συγκεκριμένος, ορισμένος από τον προγραμματιστή, αριθμός εποχών (*epochs*) στην διάρκεια των οποίων μια συγκεκριμένη μετρική του μοντέλου δεν παρουσιάσει εμφανή βελτίωση από εποχή σε εποχή. Συνήθως επιλέγεται η “παρακολούθηση” του μέσου *Validation Loss* κάθε εποχής, δηλαδή το συνολικό μέσο σφάλμα του στο σύνολο επικύρωσης, κάτι το οποίο ακολουθήθηκε και στην παρούσα εργασία, ενώ ο αριθμός των εποχών παρακολούθησης τέθηκε ίσος με 15. Επιπλέον η κλάση αυτή αρχικοποιήθηκε με τέτοιον τρόπο ώστε μετά το τέλος της εκπαίδευσης στην μνήμη να βρίσκεται το μοντέλο που εμφάνισε το χαμηλότερο *Validation Loss*, δηλαδή η έκδοση

αυτού που κατείχε τον καλύτερο και πιο αποδοτικό συνδυασμό παραμέτρων. Τέλος η τεχνική του “Πρόωρου Τερματισμού” χρησιμοποιείται γενικά ευρέως για την αποφυγή της υπερ-προσαρμογής. Συνεχίζοντας μια ακόμα τεχνική που χρησιμοποιήθηκε υλοποιείται μέσω της κλάσης *ReduceLROnPlateau*. Κατά τη συγκεκριμένη τεχνική η υπερ-παραμέτρος του ρυθμού μάθησης μειώνεται κατά έναν παράγοντα κάθε φορά που ολοκληρώνεται ένας ορισμένος αριθμός εποχών χωρίς να έχει βελτιωθεί μια μετρική του μοντέλου που παρακολουθείται και μέχρι αυτός να φτάσει μια επίσης ορισμένη κατώτερη τιμή. Γενικά η συγκεκριμένη τεχνική συνήθως βοηθάει στην εκπαίδευση των μοντέλων, “ξεκολλώντας” τα από “plateau”, κάτι το οποίο παρατηρήθηκε και κατά την εκπαίδευση του μοντέλου της εργασίας. Η μετρική τώρα που ορίστηκε να παρακολουθείται στην συγκεκριμένη εργασία ήταν παρομοίως το *Validation Loss* ενώ ο αριθμός των εποχών παρακολούθησης αυτής της μεθόδου τέθηκε ίσος με 5, ο παράγοντας μείωσης ίσος με 0.25 και η κατώτερη αποδεκτή τιμή του ρυθμού μάθησης ίση με 0.0001. Εδώ να σημειωθεί ότι ο *Adam* βελτιστοποιητής που χρησιμοποιήθηκε για την εκπαίδευση του μοντέλου ορίστηκε με αρχικό ρυθμό μάθησης 0.001, που είναι και η τυπική τιμή του.

Τέλος κλείνοντας την περιγραφή των λεπτομερειών της εκπαίδευσης του προτεινόμενου τεχνητού νευρωνικού δικτύου η εκπαίδευση έγινε ανά *batches*, δηλαδή το δίκτυο σε κάθε εποχή δεν επεξεργαζόταν όλα μαζί τα δεδομένα εισόδου αλλά το έκανε ανά σύνολα, με το μέγεθος του *batch* να ορίστηκε σε 128 παράθυρα την φορά. Επίσης η εκπαίδευση του ολοκληρώθηκε έπειτα από συνολικά 46 εποχές.

5. Πειραματική Αποτίμηση

5.1: Το σύνολο δεδομένων της εργασίας

Για την παρούσα εργασία χρησιμοποιήθηκε ένα σύνολο δεδομένων το οποίο αποτελούταν από βίντεο τα οποία επιλέχθηκαν από τρία διαφορετικά δημόσια σύνολα δεδομένων. Συγκεκριμένα συλλέχθηκαν βίντεο από το YawDD, το UTA-RLDD και ένα ακόμα σχετικά μικρό σύνολο [24]. Διαλέχθηκαν συνολικά 180 βίντεο 80 που να περιέχουν καταστάσεις εγρήγορσης και 100 που να περιέχουν διάφορες καταστάσεις υπνηλίας (χασμουρητό, μικρο-ύπνος, ύπνος κ.α.), ενώ τόσο το πλήθος των βίντεο της εγρήγορσης όσο και αυτό των βίντεο της υπνηλίας χωρίστηκε ακριβώς στην μέση σε γυναίκες και άντρες. Παράλληλα ο ρυθμός ανανέωσης των καρτέ κάθε βίντεο ήταν περίπου 30. Επιπρόσθετα για να διασφαλιστεί ότι τα δεδομένα θα διακατέχονται από ποικιλομορφία με στόχο την ανάπτυξη ενός στιβαρού μοντέλου που θα είναι σε θέση να γενικεύει καλύτερα και πιο αποτελεσματικά επιλέχθηκαν βίντεο με άτομα διαφορετικών ηλικιακών ομάδων και εμφανισιακών χαρακτηριστικών, τόσο με γυαλιά οράσεως αλλά και χωρίς. Στην συνέχεια φαίνεται ένα σχεδιάγραμμα που κατασκευάστηκε με την χρήση της βιβλιοθήκης Matplotlib, μιας Python βιβλιοθήκης που χρησιμοποιείται ευρέως για οπτικοποιήσεις δεδομένων, και δείχνει την γενική μορφή των βίντεο του dataset παρουσιάζοντας το πρώτο καρτέ από 6 χειροκίνητα επιλεγμένα βίντεο αυτού.



Εικόνα 21: Παράδειγμα Βίντεο Dataset

5.2: Πληροφορίες διαχωρισμού dataset & δημιουργίας παραθύρων

Όπως έχει αναφερθεί ήδη σε προηγούμενο κεφάλαιο το αρχικό σύνολο δεδομένων χωρίστηκε σε σύνολα εκπαίδευσης, επικύρωσης και αξιολόγησης με την βοήθεια μιας συνάρτησης που φτιάχτηκε για αυτόν τον σκοπό και κλήθηκε 2 φορές. Ειδικότερα το αρχικό σύνολο διασπάστηκε σε ένα υποσύνολο με μέγεθος περίπου το 20% του αρχικού το οποίο αποτέλεσε και το σύνολο αξιολόγησης (testing) ενώ το σύνολο που αναλογούσε στο υπόλοιπο ποσοστό διασπάστηκε με την δεύτερη κλήση της συνάρτησης πάλι σε μια αναλογία περίπου 80-20 σε σύνολα εκπαίδευσης και επικύρωσης αντίστοιχα. Σε ότι αφορά την δημιουργία των χρονικών παραθύρων, των ετικετών αυτών καθώς και την συλλογή του αναγνωριστικού του βίντεο στο οποίο ανήκε το κάθε παράθυρο αυτό έγινε με την χρήση συνάρτησης που έχει περιγραφεί και κλήθηκε 3 φορές, μια για κάθε σύνολο δεδομένων του προηγούμενου βήματος. Περνώντας σε περισσότερες λεπτομέρειες ως μέγεθος των παραθύρων ορίστηκαν τα 30 καρέ, δοκιμάστηκαν και τα 20 οδήγησαν όμως σε χειρότερη απόδοση το μοντέλο, και το βήμα για την δημιουργία των παραθύρων καθορίστηκε στα 2 καρέ, ελέγχθηκαν επίσης τα 3 και το 1 αλλά παρομοίως προκάλεσαν πτώση των επιδόσεων του δικτύου. Στην εικόνα που ακολουθεί, που προέρχεται από το τερματικό της εφαρμογής “Visual Studio Code” που χρησιμοποιήθηκε για την γραφή των προγραμμάτων, φαίνονται το ποσοστό των δεδομένων του dataset που περιέγραφαν υπνηλία, οι διαχωρισμοί αυτού που πραγματοποιήθηκαν όπως και τα αντίστοιχα ποσοστά τους αλλά και το ποσοστό των δεδομένων υπνηλίας κάθε υποσυνόλου, ακόμα είναι εμφανείς οι διαστάσεις όλων των συνόλων των παραθύρων, των ετικετών και των αναγνωριστικών που παράχθηκαν. Να σημειωθεί ότι σύνολο με αναγνωριστικά δημιουργήθηκε και για το σύνολο δεδομένων εκπαίδευσης δεν τυπώθηκε όμως στο τερματικό η διάσταση του διότι δεν χρησιμοποιήθηκε το συγκεκριμένο σύνολο σε καμία φάση της εργασίας.

```
Dataset drowsiness ratio: 30.55%
Found an acceptable split!
Test dataset size: 0.1902 | Test dataset drowsiness ratio: 0.2916
Train dataset size: 0.8098 | Train dataset drowsiness ratio: 0.3087
Found an acceptable split!
Validation dataset size: 0.1962 | Validation dataset drowsiness ratio: 0.2936
Final train dataset size: 0.8038 | Final train dataset drowsiness ratio: 0.3124
Creation of the windows finished successfully!
Train features dataset shape: (3981, 30, 2) | Train labels dataset shape: (3981,)
Test features dataset shape: (1140, 30, 2) | Test labels dataset shape: (1140,)
Validation features dataset shape: (951, 30, 2) | Validation labels dataset shape: (951,)
Test id's shape: (1140,) | Validation id's shape: (951,)
=====
```

Εικόνα 22: Σύνολα που Παράχθηκαν κατά την Προεπεξεργασία

5.3: Μετρικές που χρησιμοποιήθηκαν

Η επιλογή των σωστών μετρικών κατά την αξιολόγηση ενός μοντέλου είναι καίριας σημασίας. Οι πιο γνωστές και ευρέως χρησιμοποιούμενες μετρικές για την αξιολόγηση μοντέλων που εκτελούν ταξινόμηση, είτε δυαδική είτε πολυταξική, είναι οι *Accuracy*, *Precision*, *Recall* και *F1-Score*. Αυτές ακριβώς οι μετρικές χρησιμοποιήθηκαν και στην παρούσα εργασία που εκτελείται δυαδική ταξινόμηση με την κάθε μια να έχει την δική της ερμηνεία και σημαντικότητα. Για τον υπολογισμό από την άλλη αυτών των τιμών χρησιμοποιούνται οι προβλέψεις του μοντέλου αφού πρώτα μετατραπούν σε ετικέτες από πιθανότητες με βάση ένα όριο και συγκριθούν κατόπιν με τις πραγματικές ετικέτες των δεδομένων. Ειδικότερα μετά το πέρας της σύγκρισης οι προβλέψεις χωρίζονται σε 4 κατηγορίες αναλόγως με το αν ήταν σωστές και το σε ποια κλάση αναφέρονται οι οποίες είναι οι Αληθώς Θετικό (TP), Αληθώς Αρνητικό (TN), Ψευδώς Θετικό (FP) και Ψευδώς Αρνητικό (FN). Αναλυτικότερα ως θετική ορίζεται στην δυαδική ταξινόμηση η κλάση 1, στην παρούσα εργασία η υπνηλία, ενώ ως αρνητική ορίζεται η κλάση 0, η εγρήγορση. Όλα αυτά οπτικοποιούνται με την χρήση ενός Πίνακα Σύγχυσης (Confusion Matrix). Παρακάτω δίνεται ένα παράδειγμα της μορφής που μπορεί να έχει ένας πίνακας σύγχυσης σε ένα πρόβλημα δυαδικής ταξινόμησης και περιγράφονται συνοπτικά οι προαναφερθέντες μετρικές και ο τρόπος υπολογισμού τους με βάση αυτές τις κατηγορίες.

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Εικόνα 23: Παράδειγμα Μορφής Πίνακα Σύγχυσης

Accuracy:

Αποτελεί την πιο απλή και κατανοητή μετρική. Επί της ουσίας ερμηνεύεται ως εξής “Από όλες τις προβλέψεις που πραγματοποίησε το μοντέλο και για τις 2 κλάσεις πόσες ήταν σωστές”. Ο τύπος της είναι ο ακόλουθος:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

Εξίσωση 15: Μαθηματικός Τύπος Accuracy

Είναι μια ιδιαίτερα χρήσιμη μετρική από μόνη της όμως δεν παρουσιάζει μια ολοκληρωμένη και αντικειμενική εικόνα της απόδοσης του μοντέλου ειδικά όταν το σύνολο δεδομένων εκπαίδευσης είναι μη ισορροπημένο ανάμεσα στις 2 κλάσεις. Ειδικότερα αν ένα μοντέλο πραγματοποιεί συνεχώς σωστές προβλέψεις για τα δείγματα της πλειοψηφικής-αρνητικής κλάσης που είναι πολλαπλάσια έναντι αυτών της μειοψηφικής-θετικής, τα δείγματα της οποίας δυσκολεύεται να ταξινομήσει σωστά, θα έχει πολύ υψηλή ακρίβεια αλλά δεν θα είναι χρήσιμο στο ζητούμενο του προβλήματος που είναι κυρίως η σωστή ταξινόμηση της κλάσης 1, στην περίπτωση μας της υπνηλίας.

Precision:

Μια ακόμα χρήσιμη μετρική η οποία περιγράφεται ως εξής “Από όλες τις προβλέψεις που αφορούσαν την θετική κλάση πόσες ήταν σωστές”. Παρόμοιος είναι ο ορισμός της μετρικής αυτής και για την αρνητική κλάση ενώ ο τύπος της για την θετική κλάση είναι ο ακόλουθος:

$$\text{Precision} = TP / (TP + FP)$$

Εξίσωση 16: Μαθηματικός Τύπος Precision

Υψηλό precision σημαίνει πως το μοντέλο παράγει λίγα FP, δηλαδή λίγες φορές θα προβλέψει ότι η κλάση είναι 1 ενώ στην πράξη ήταν 0. Είναι ιδιαίτερα σημαντική μετρική επομένως σε προβλήματα που οι λίγες ψευδώς θετικές προβλέψεις είναι καίριας σημασίας.

Recall:

Ερμηνεύεται ως εξής για την θετική κλάση “Από όλα τα δείγματα που έπρεπε να προβλεφθούν ως θετικά πόσα κατάφερε να αναγνωρίσει το μοντέλο”, ενώ ο ορισμός της για την αρνητική κλάση είναι παρόμοιος. Ο τύπος της για την κλάση 1 δίνεται στην συνέχεια:

$$\text{Recall} = TP / (TP + FN)$$

Εξίσωση 17: Μαθηματικός Τύπος Recall

Υψηλό recall σημαίνει πως το μοντέλο σπάνια δεν καταφέρνει να αναγνωρίσει ένα θετικό δείγμα ως τέτοιο, επομένως διακατέχεται από υψηλή σημασία σε προβλήματα που είναι κρίσιμο η θετική κλάση να αναγνωρίζεται όσο το δυνατόν πιο αποτελεσματικά. Ένα τέτοιο πρόβλημα είναι και η αναγνώριση της υπνηλίας κατά την οδήγηση. Γενικά το precision και το recall βρίσκονται σε “ανταγωνισμό” μεταξύ τους καθώς σε ορισμένες περιπτώσεις υψηλό recall μπορεί παράλληλα να σημαίνει χαμηλό precision, το μοντέλο εκτελεί “επιθετική” αναγνώριση της κλάσης 1, ή αντίστοιχα υψηλό precision μπορεί να οδηγεί σε χαμηλό recall, πραγματοποιεί προβλέψεις για την κλάση 1 μόνο όταν είναι πολύ “σίγουρο”.

F1-Score:

Η συγκεκριμένη μετρική αποτελεί έναν δείκτη ισορροπίας ανάμεσα στο *precision* και το *recall* παρουσιάζοντας την γενική απόδοση του μοντέλου στην εργασία της ταξινόμησης. Ο τύπος της είναι ο ακόλουθος:

$$F1 = 2 \times (Precision \times Recall) / (Precision + Recall)$$

Εξίσωση 18: Μαθηματικός Τύπος F1-Score

Μπορεί να πάρει υψηλές τιμές μόνο αν και οι δύο διαφορετικές μετρικές στον τύπο της έχουν υψηλές τιμές. Αποτελεί πολύ χρήσιμη μετρική όταν σε ένα πρόβλημα δεν υπάρχει σημαντικός λόγος να υπεριοχύνει ιδιαίτερα το *recall* έναντι του *precision*, και το αντίστροφο, αλλά και οι δύο μετρικές έχουν περίπου την ίδια σημασία.

5.4: Εύρεση ορίου πιθανοτήτων & εξομάλυνση αυτών

Για την μετατροπή όλων των πιθανοτήτων που παρήγαγε το μοντέλο σε ετικέτες, κατά την διαδικασία της αξιολόγησης, ώστε μετά να υπολογιστούν οι ανάλογες μετρικές χρησιμοποιήθηκε συγκεκριμένο όριο, που βρέθηκε με την διαδικασία που θα αναφερθεί στην συνέχεια, και όχι το τυπικό όριο του 0.5 που χρησιμοποιείται στις περισσότερες βιβλιοθήκες, δηλαδή πιθανότητες ≥ 0.5 κατατάσσονται στην θετική κλάση αλλιώς στην αρνητική. Αυτό έγινε για να μετατραπούν πιο αποτελεσματικά και σωστά οι πιθανότητες σε ετικέτες και κατά συνέπεια να βελτιωθούν οι μετρικές του μοντέλου. Ακόμα υλοποιήθηκε μια συνάρτηση η οποία εφαρμόστηκε σε όλες τις εξαγόμενες πιθανότητες του μοντέλου, πριν μετατραπούν σε ετικέτες, πραγματοποιώντας “εξομάλυνση” αυτών με στόχο την μείωση μεμονωμένων-στιγμιαίων λαθών (FP, FN). Η συνάρτηση αυτή λοιπόν δέχεται ως παραμέτρους το σύνολο των προβλέψεων και αυτό των αναγνωριστικών των βίντεο στα οποία ανήκουν τα παράθυρα για τα οποία πραγματοποιήθηκαν οι προβλέψεις, όπως και μια σταθερά που καθορίζει το μέγεθος του κυλιόμενου παραθύρου “εξομάλυνσης”, περισσότερα για αυτό στην συνέχεια. Φτιάχτηκε επομένως ένα *pandas Dataframe* με βάση τα αναγνωριστικά και τις προβλέψεις και μέσα σε έναν βρόγχο πραγματοποιήθηκε η “εξομάλυνση” των πιθανοτήτων για κάθε βίντεο ξεχωριστά εκμεταλλευόμενοι τα αναγνωριστικά των αυτών, η διαδικασία ήταν απαραίτητο να γίνει για κάθε βίντεο ξεχωριστά διότι άλλα βίντεο περιέγραφαν υπνηλία και άλλα εγρήγορση. Η εξομάλυνση έγινε με την χρήση ενός κυλιόμενου παραθύρου σταθερού μεγέθους και βήματος μιας πιθανότητας το οποίο εφαρμόζοταν στις πιθανότητες κάθε βίντεο αντικαθιστώντας την πιο δεξιά πιθανότητα στο παράθυρο με το μέσο του (*median*), λαμβάνονταν δηλαδή υπόψιν οι παρελθοντικές πιθανότητες για την “εξομάλυνση”. Με αυτό τον τρόπο λαμβανόταν υπόψιν το συνολικό “περιβάλλον” κάθε πιθανότητας, για παράδειγμα αν ένα βίντεο που αφορούσε εγρήγορση, άρα όλες οι πιθανότητες για αυτό θα πρέπει να έχουν χαμηλές τιμές, εμφανίσει στο σύνολο των πιθανοτήτων του μια υψηλή τιμή αυτή με την προαναφερθείσα τεχνική θα διορθωνόταν ως Ψευδώς Θετική σε μια χαμηλή τιμή. Τώρα σε ότι αφορά το μέγεθος αυτού του παραθύρου που περάστηκε ως

παράμετρος στην συνάρτηση αυτό καθορίστηκε μαζί με το όριο των πιθανοτήτων, που αναφέρθηκε παραπάνω, με την ακόλουθη μέθοδο. Αρχικά πραγματοποιήθηκαν προβλέψεις πάνω στο Validation Set. Έπειτα δημιουργήθηκε ένας διπλός βρόγχος, εμπνευσμένος από την τεχνική “Grid Search” που χρησιμοποιείται συχνά στην μηχανική μάθηση για την εύρεση του βέλτιστου συνδυασμού υπερ-παραμέτρων, με τον εξωτερικό βρόγχο να “ελέγχει” πιθανές τιμές μεγέθους για το παράθυρο “εξομάλυνσης” και τον εσωτερικό για κάθε μια από αυτές τις τιμές να “ελέγχει” διαφορετικά όρια από ένα σύνολο. Επί της ουσίας για κάθε μέγεθος παραθύρου γινόταν εξομάλυνση των πιθανοτήτων με αυτό και έπειτα στον εσωτερικό βρόγχο μετατρέπονταν αυτές οι πιθανότητες σε ετικέτες κάθε φορά με ένα διαφορετικό όριο το οποίο αντλούνταν από ένα σύνολο και στην συνέχεια υπολογιζόταν το F1-Score ανάμεσα στις προβλεπόμενες και τις πραγματικές ετικέτες. Ειδικότερα το σύνολο των μεγεθών περιείχε τις τιμές [1, 3, 5, 7, 9, 11, 13] ενώ αυτό των ορίων αριθμούς στο διάστημα [0.3, 0.7] με τον κάθε αριθμό να διαφέρει από τον προηγούμενο 0.002. Περνώντας τώρα στις λεπτομέρειες επιλογής του καλύτερου ζεύγους μεγέθους/ορίου κάθε συνδυασμός μέγεθος/όριο/F1-Score αποθηκευόταν ως Python Dictionary σε μια Python λίστα. Η λίστα αυτή μετατράπηκε σε Pandas Dataframe και βρέθηκε το υψηλότερο F1-Score ανάμεσα σε όλους τους πιθανούς συνδυασμούς. Έπειτα το Dataframe “φιλτραρίστηκε” ώστε να κρατηθούν όλοι οι συνδυασμοί των οποίων το F1-Score ήταν πολύ κοντά στο μέγιστο, για την ακρίβεια κατά μέγιστο 0.005 μικρότερο από αυτό. Κατόπιν αναζητήθηκε το ελάχιστο μέγεθος παραθύρου από τους παραπάνω συνδυασμούς και απομονώθηκαν από αυτούς τα όρια που χρησιμοποιήθηκαν με αυτό το μέγεθος. Εν τέλη τον ιδανικό συνδυασμό μεγέθους/ορίου αποτέλεσαν το ελάχιστο μέγεθος και η μέση τιμή όλων των ορίων που αναφέρθηκαν παραπάνω. Το ελάχιστο ιδανικό μέγεθος παραθύρου επιλέχθηκε ώστε να μειωθεί όσο το δυνατόν η καθυστέρηση στις προβλέψεις του μοντέλου που θα εισάγει η εξομάλυνση κατά την εκτέλεση σε live περιβάλλον ενώ η μέση τιμή των ιδανικών ορίων επιλέχθηκε με στόχο να γίνει το μοντέλο πιο στιβαρό και ανθεκτικό σε μικρές μεταβολές των πιθανοτήτων που μπορούν να εμφανιστούν σε ένα τέτοιο περιβάλλον. Οι τιμές τώρα των σταθερών αυτών που χρησιμοποιήθηκαν εν τέλη στις πιθανότητες του προτεινόμενου μοντέλου θα παρουσιαστεί στο επόμενο κεφάλαιο.

5.5: Πειραματισμός & τελικό προτεινόμενο σύστημα

Στα πλαίσια πειραματισμού πραγματοποιήθηκε μια σύγκριση των μετρικών που προέκυψαν για το μοντέλο της εργασίας με την δοκιμή διαφορετικών ρυθμών επεξεργασίας δεδομένων έχοντας διατηρήσει όλες τις υπόλοιπες παραμέτρους του συνόλου των προγραμμάτων των σταδίων της εργασίας ίδιες και σταθερές. Άρα ο πειραματισμός, για να είναι αντικειμενικός, έγινε με μόνο παράγοντα διαφοροποίησης και σύγκρισης των διαφορετικών μετρικών αυτόν τον ρυθμό. Συγκεκριμένα, όπως είχε αναφερθεί στην ενότητα 4.1.2, κατά την προ-επεξεργασία των αρχικών δεδομένων και την δημιουργία του αρχείου κειμένου μπορεί να “διαβαζόταν” κάθε καρέ καθενός διαθέσιμου βίντεο επεξεργάζονταν όμως τελικώς, για να υπολογιστούν οι μαθηματικοί λόγοι που παρουσιάστηκαν και να δημιουργηθεί το αρχείο κειμένου, καρέ με έναν χαμηλότερο ρυθμό από ότι “διαβάζονταν”, συγκεκριμένα 1 στα 5 (6fps) ενώ τα βίντεο ήταν στα περίπου 30fps. Γιατί τελικώς επιλέχθηκαν τα 6fps έναντι άλλων τιμών που

δοκιμάστηκαν γίνεται φανερό από τις εικόνες των μετρικών που προέκυψαν, ύστερα από την εκπαίδευση και αξιολόγηση του μοντέλου με βάση τα διαφορετικά σύνολα δεδομένων-αρχεία κειμένου που έδωσε ο κάθε ρυθμός, για κάθε ρυθμό και ακολουθούν στην συνέχεια. Ακόμα σημαντικό είναι να σημειωθεί ότι μέσω κατάλληλης εντολής όλες οι γεννήτριες ψευδο-τυχαίων αριθμών που χρησιμοποιούνται κατά την ανάπτυξη και εκπαίδευση ενός τεχνητού νευρωνικού δικτύου ρυθμίστηκαν με τέτοιον τρόπο ώστε κάθε φορά που εκτελείται η εκπαίδευση του μοντέλου να παράγουν τους ίδιους ακριβώς αριθμούς, οπότε αυτό απαιτείται, παρόμοια και όλες οι διεργασίες/αλγόριθμοι που χρησιμοποιεί η βιβλιοθήκη TensorFlow μέσω εντολής ορίστηκαν να είναι όσο γίνεται πιο ντετερμινιστικοί. Αυτό έγινε για να ενισχυθεί η αντικειμενικότητα κατά τον πειραματισμό τόσο μεταξύ διαφορετικών ρυθμών επεξεργασίας καρέ όσο και μεταξύ αλλαγών που μπορεί να έγιναν στην ίδια την αρχιτεκτονική-εκπαίδευση του μοντέλου, εξαλείφοντας το στοιχείο της τυχαιότητας, αφού με αυτές τις ρυθμίσεις το μοντέλο παράγει κάθε φορά τα ίδια ακριβώς αποτελέσματα/μετρικές, οπότε εκτελείται ο κώδικας του, αν δεν πραγματοποιηθεί κάποια αλλαγή σε αυτόν.

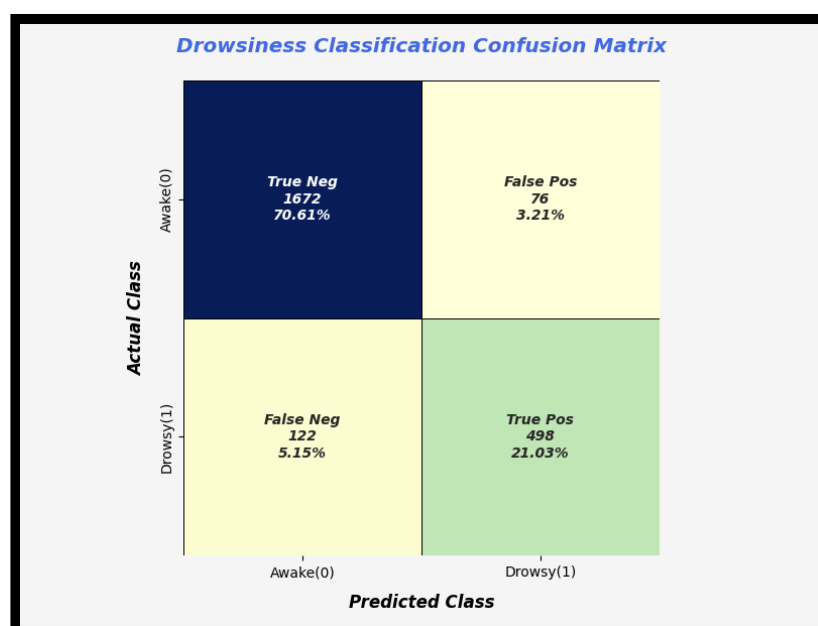
```

=====
Model's loss during evaluation: 0.0225
=====
16/16 ----- 1s 35ms/step
=====
Best threshold found: 0.3620 | Best rolling window size found: 11
=====
19/19 ----- 0s 7ms/step
Classification Report:
=====

```

	precision	recall	f1-score	support
Awake(0)	0.93	0.96	0.94	1748
Drowsy(1)	0.87	0.80	0.83	620
accuracy			0.92	2368
macro avg	0.90	0.88	0.89	2368
weighted avg	0.92	0.92	0.92	2368

Εικόνα 24: Μετρικές Μοντέλου για τα 15fps



Εικόνα 25: Πίνακας Σύγχυσης Μοντέλου για τα 15fps

Εδώ να σημειωθεί ότι τα ποσοστά στον πίνακα σύγχυσης στο κάθε κελί αυτού υποδηλώνουν το ποσοστό που καταλαμβάνουν οι προβλέψεις της εκάστοτε κατηγορίας έναντι του συνολικού αριθμού των προβλέψεων που εκτέλεσε το μοντέλο κατά την διαδικασία της αξιολόγησης. Για παράδειγμα οι Ψευδώς Θετικές προβλέψεις αποτέλεσαν το 3.21% του συνολικού αριθμού των προβλέψεων από όλες τις κατηγορίες. Ακολουθούν οι μετρικές και για τους υπόλοιπους ρυθμούς επεξεργασίας:

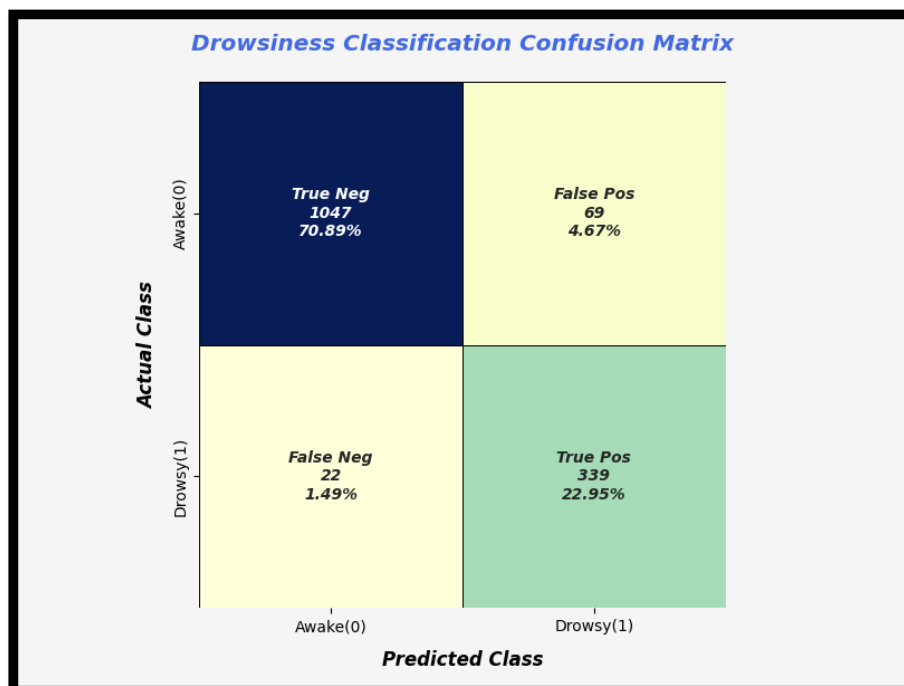
```

=====
Model's loss during evaluation: 0.0202
=====
10/10 1s 71ms/step
=====
Best threshold found: 0.3130 | Best rolling window size found: 13
=====
12/12 0s 9ms/step
Classification Report:
=====

```

	precision	recall	f1-score	support
Awake(0)	0.98	0.94	0.96	1116
Drowsy(1)	0.83	0.94	0.88	361
accuracy			0.94	1477
macro avg	0.91	0.94	0.92	1477
weighted avg	0.94	0.94	0.94	1477

Εικόνα 26: Μετρικές Μοντέλου για τα 10 fps



Εικόνα 27: Πίνακας Σύγχυσης Μοντέλου για τα 10fps

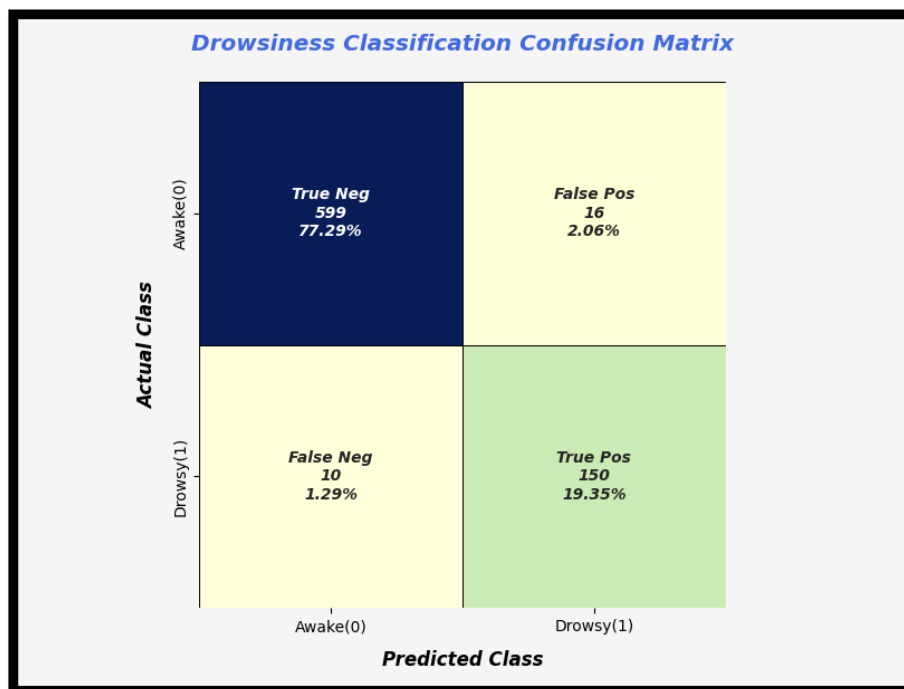
```

=====
Model's loss during evaluation: 0.0168
=====
6/6 1s 117ms/step
=====
Best threshold found: 0.4379 | Best rolling window size found: 9
=====
7/7 0s 10ms/step
Classification Report:

```

	precision	recall	f1-score	support
Awake(0)	0.98	0.97	0.98	615
Drowsy(1)	0.90	0.94	0.92	160
accuracy			0.97	775
macro avg	0.94	0.96	0.95	775
weighted avg	0.97	0.97	0.97	775

Εικόνα 28: Μετρικές Μοντέλου για τα 6fps



Εικόνα 29: Πίνακας Σύγχυσης Μοντέλου για τα 6fps

Όπως φαίνεται και από τις εικόνες τελικώς τα 6fps υπερίσχυσαν έναντι των άλλων τιμών που δοκιμάστηκαν καθώς όταν το μοντέλο εκπαιδεύτηκε και αξιολογήθηκε με το dataset που αυτός ο ρυθμός έδωσε κατάφερε και υψηλότερες τιμές να επιτύχει σχεδόν σε όλες τις μετρικές του Classification Report και να σημειώσει τα λιγότερα σφάλματα (FP, FN) έναντι των “ανταγωνιστικών” ρυθμών. Εδώ να σημειωθεί ότι η σύγκριση των μετρικών που παρουσιάστηκε πραγματοποιήθηκε με την δημιουργία των χρονικών παραθύρων για το μοντέλο σε κάθε περίπτωση να γίνεται με το ίδιο βήμα (stride) με τιμή 3 καρέ. Τελικώς ο ρυθμός που διατηρήθηκε είναι, φυσικά, τα 6fps όμως η δημιουργία των παραθύρων αποφασίστηκε να πραγματοποιείται με βήμα ίσο

με 2 καρτέ, καθώς δοκιμάστηκε και βελτιώσε ακόμα περισσότερο την απόδοση του προτεινόμενου μοντέλου. Οι τελικές μετρικές που προκύπτουν για το μοντέλο είναι οι ακόλουθες:

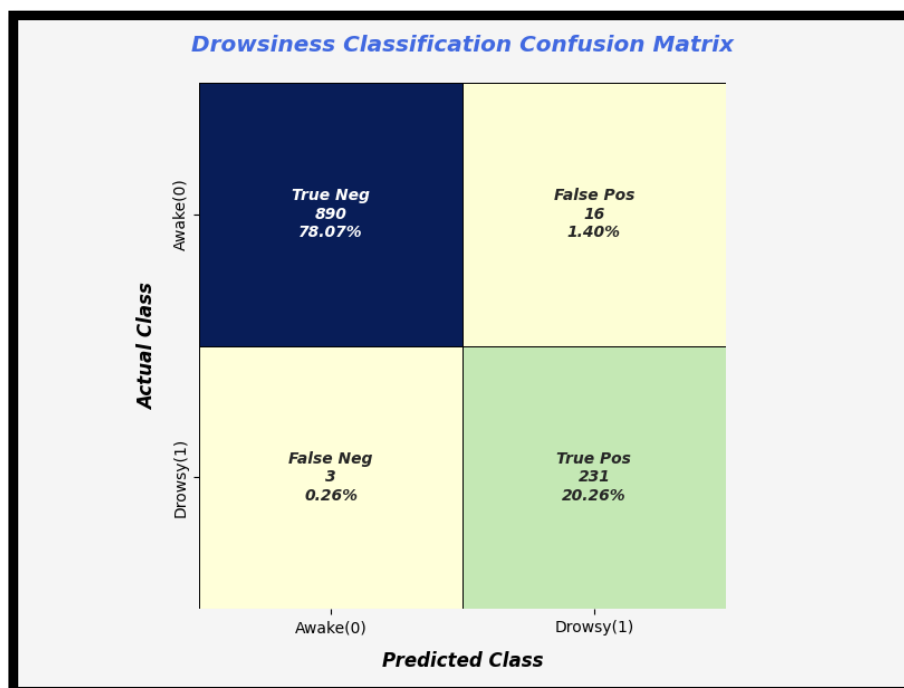
```

=====
Model's loss during evaluation: 0.0151
=====
8/8 1s 68ms/step
=====
Best threshold found: 0.3753 | Best rolling window size found: 13
=====
9/9 0s 8ms/step
Classification Report:
=====

```

	precision	recall	f1-score	support
Awake(0)	1.00	0.98	0.99	906
Drowsy(1)	0.94	0.99	0.96	234
accuracy			0.98	1140
macro avg	0.97	0.98	0.97	1140
weighted avg	0.98	0.98	0.98	1140

Εικόνα 30: Μετρικές Μοντέλου για τα 6fps και Βήμα 2 Καρέ



Εικόνα 31: Πίνακας Σύγχυσης Μοντέλου για τα 6fps και Βήμα 2 Καρέ

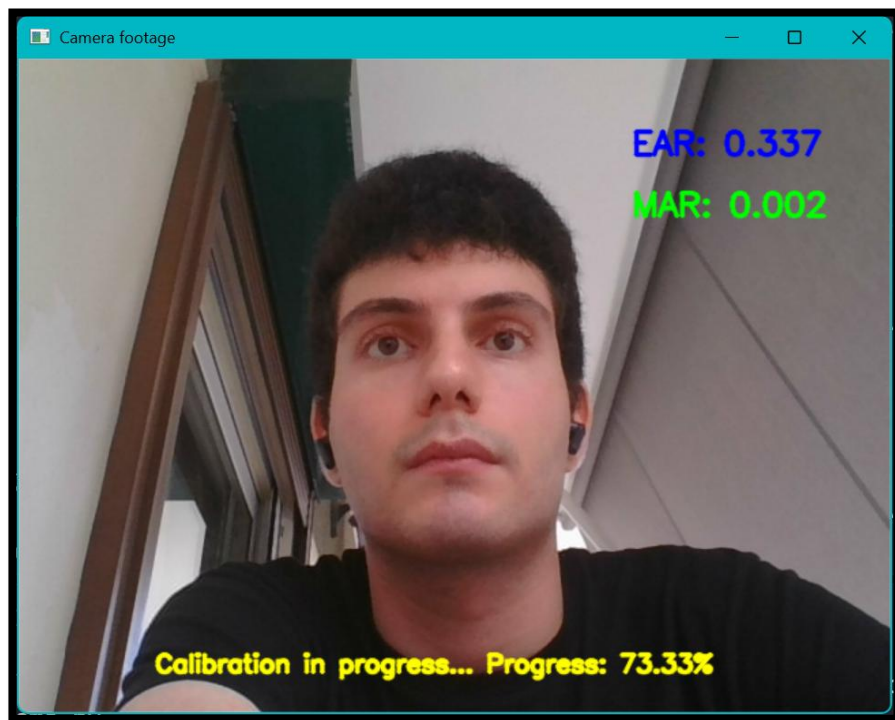
Από τις παραπάνω εικόνες εξάγονται τα εξής, το μέγεθος του παραθύρου εξομάλυνσης προέκυψε 13, το όριο για την μετατροπή των πιθανοτήτων 0.3753 ενώ όλες οι μετρικές του μοντέλου “αγγίζουν” πολύ υψηλά επίπεδα και παράλληλα οι εσφαλμένες προβλέψεις του είναι σημαντικά λίγες.

5.6: Παρουσίαση live demo εργασίας

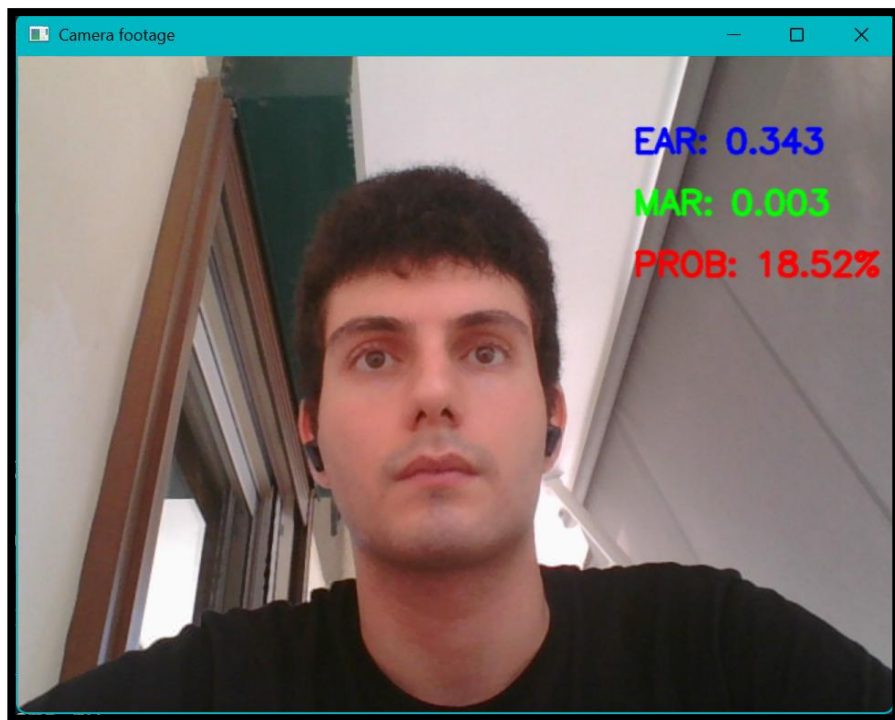
Στα πλαίσια της εργασίας δημιουργήθηκε ένα live demo που συνδυάζει τεχνικές που αναφέρθηκαν σε προηγούμενα στάδια αυτής για να μπορεί με την χρήση του προτεινόμενου μοντέλου να προβλέπει αξιόλογα σε ζωντανό χρόνο την κατάσταση ενός ατόμου που στέκεται μπροστά από την κάμερα του υπολογιστή. Ο τρόπος που οργανώθηκε και δημιουργήθηκε αυτό το demo έχει ως εξής. Αρχικά δημιουργήθηκε ένας ατέρμονος βρόγχος έτσι ώστε με την χρήση της βιβλιοθήκης OpenCV να “διαβάζονται” συνεχώς καρέ από την κάμερα μέχρις ότου ο χρήστης να πατήσει το πλήκτρο “q” του υπολογιστή όπου σε αυτήν την περίπτωση σταματούσε η “ανάγνωση” των καρέ και κατά συνέπεια η εκτέλεση του demo. Τα καρέ που διαβάζονταν τώρα στέλνονταν για επεξεργασία στα μοντέλα της MediaPipe με ρυθμό επεξεργασίας 1 στα 5 (6fps), ακριβώς δηλαδή με τον ίδιο ρυθμό που επιλέχθηκε να επεξεργάζονται για να δημιουργηθεί το αρχείο κειμένου κατά την φάση προ-επεξεργασίας των δεδομένων της εργασίας. Το task από την άλλη της MediaPipe που αναλαμβάνει αυτήν την επεξεργασία αρχικοποιήθηκε και δημιουργήθηκε με τέτοιον τρόπο ώστε να αφορά εκτέλεση σε live περιβάλλον. Αυτό έγινε διότι αν αρχικοποιηθεί με αυτόν τον τρόπο όλη η διεργασία υπολογισμού των σημείων-οροσήμων εκτελείται σε ξεχωριστό νήμα από αυτό που δεσμεύτηκε για το βασικό πρόγραμμα-διεργασία με αποτέλεσμα να μην καθυστερείται η εκτέλεση των υπόλοιπων λειτουργιών του προγράμματος, κυρίως δηλαδή η εκτέλεση του βρόγχου “ανάγνωσης” της κάμερας. Αυτό αποτελεί ένα πολύ σημαντικό χαρακτηριστικό όταν ένα πρόγραμμα εκτελείται σε “ζωντανό περιβάλλον” όπου η ελάττωση της καθυστέρησης, ειδικά στην προκειμένη περίπτωση που στόχος είναι η αναγνώριση της υπνηλίας, όσο γίνεται πιο πολύ έχει πολύ μεγάλη σημασία. Επομένως για τον εντοπισμό και υπολογισμό των σημείων από την MediaPipe, όποτε αυτό έπρεπε να γίνει, πραγματοποιούνταν κάθε φορά κλήση μιας ασύγχρονης συνάρτησης. Στην συνέχεια όταν η MediaPipe θα είχε υπολογίσει τα σημεία του εκάστοτε καρέ πραγματοποιούσε κλήση μιας συγκεκριμένης callback συνάρτησης, με βάση το αρι της, στην οποία εντάχθηκε όλος ο κώδικας της διαδικασίας που ακολουθεί τον εντοπισμό των οροσήμων. Η συγκεκριμένη συνάρτηση εκτελείται επίσης στο νήμα που δεσμεύτηκε για την MediaPipe. Περνώντας τώρα σε λεπτομέρειες σχετικά με την διαδικασία που ακολούθησε τον εντοπισμό των σημείων αρχικά για την διαχείριση των διανυσμάτων 2 τιμών [EAR, MAR] καθενός καρέ καθώς και για την εξομάλυνση των πιθανοτήτων με χρήση ενός “παραθύρου” 13 τιμών, όσο επιλέχθηκε δηλαδή τελικώς το μέγεθος με βάση το κεφάλαιο 5.5, χρησιμοποιήθηκε η δομή δεδομένων deque της βιβλιοθήκης collections. Συγκεκριμένα η δομή αυτή είναι ένας τύπος Python λίστας που συνδυάζει τόσο τις ιδιότητες μιας ουράς (FIFO) όσο και μιας στοίβας (LIFO) επιτρέποντας γρήγορη εισαγωγή/αφαίρεση στοιχείων τόσο από την αρχή όσο και από το τέλος της καθιστώντας την ιδιαίτερα αποδοτική και χρήσιμη σε “σενάρια” που απαιτούν χρήση κυλιόμενων παραθύρων σε πραγματικό/“ζωντανό” χρόνο. Συνεπώς χρησιμοποιήθηκαν 3 διαφορετικές δομές τύπου deque, μια με μέγιστο μέγεθος 13 για τις πιθανότητες του μοντέλου, μια με μέγιστο μέγεθος 30 για το παράθυρο χαρακτηριστικών με βάση το οποίο κάθε φορά παρήγαγε μια πιθανότητα το δίκτυο και τέλος μια τρίτη με μέγιστο μέγεθος 180 διανύσματα/καρέ που διαδραμάτισε τον ρόλο ιστορικού διανυσμάτων για να μπορέσει να γίνεται αποτελεσματικά κάθε φορά η κανονικοποίηση των τιμών των λόγων EAR και MAR όλων των διανυσμάτων του

παραθύρου χαρακτηριστικών πριν αυτό “δοθεί” για επεξεργασία στο μοντέλο. Ειδικότερα κάθε φορά τον υπολογισμό των οροσήμων του εκάστοτε καρέ ακολουθούσε η αποτίμηση της μέσης τιμής του *EAR* και της μοναδικής τιμής του *MAR*, με τον τρόπο που εξηγήθηκε σε προηγούμενο κεφάλαιο, έπειτα 2 καθολικές μεταβλητές του προγράμματος δέχονταν αυτές τις τιμές και στην συνέχεια το αντίστοιχο διάνυσμα [*EAR*, *MAR*] εισερχόταν στα δεξιά της *deque* (*append*) του παραθύρου αλλά και του ιστορικού ενώ παράλληλα ένας μετρητής για την παρακολούθηση του βήματος (*stride*) αυξανόταν κατά 1. Εδώ να σημειωθεί πως η δομή δεδομένων *deque* είναι σχεδιασμένη με τέτοιο τρόπο ώστε από την στιγμή που γεμίσει με όσα στοιχεία είναι ορισμένο το μέγιστο μέγεθος της κάθε νέα εισαγωγή στα δεξιά της προκαλεί την αφαίρεση του πιο αριστερά στοιχείου, στην περίπτωση κυλιόμενου παραθύρου του παλαιότερου στοιχείου. Με αυτόν τον τρόπο υλοποιείται η λογική ενός κυλιόμενου παραθύρου κατά την εκτέλεση σε ζωντανό περιβάλλον. Συνεχίζοντας την ανάλυση πριν πραγματοποιηθεί οποιαδήποτε πρόβλεψη ελεγχόταν αν το μέγεθος της *deque* με το ιστορικό ήταν μικρότερο από 60 καρέ. Αυτό είχε σαν αποτέλεσμα για 10sec (60 καρέ / 6fps), από την στιγμή που άνοιγε η κάμερα, να πραγματοποιούνταν απλά μια σύντομη συλλογή δεδομένων με σκοπό την έγκυρη και ποιοτική κανονικοποίηση, των μαθηματικών λόγων που χρησιμοποιούνται για την αξιολόγηση της κατάστασης του ατόμου, στην συνέχεια με βάση την ελάχιστη και την μέγιστη τιμή τόσο του *EAR* όσο και του *MAR* του χρήστη. Αν τώρα το μέγεθος του ιστορικού ήταν μεγαλύτερο ή ίσο από 60 και παράλληλα ο μετρητής του βήματος ήταν μεγαλύτερος ή ίσος από το 2, με σκοπό να πραγματοποιείται πρόβλεψη από το μοντέλο κάθε 2 καρέ (περίπου 0.3sec στα 6fps) έπειτα από την πρώτη πρόβλεψη, όσο επιλέχθηκε δηλαδή να είναι και το βήμα δημιουργίας των παραθύρων όπως αναφέρεται στο κεφάλαιο 5.5, ο μετρητής μηδενιζόταν, λάμβανε “χώρα” η κανονικοποίηση των δεδομένων του παραθύρου και έπειτα η τροφοδοσία του στο μοντέλο. Συγκεκριμένα η διαδικασία της κανονικοποίησης υλοποιήθηκε ακολουθώντας την λογική που παρουσιάστηκε στο κεφάλαιο 4.2.3. Από την άλλη με σκοπό την ελαχιστοποίηση της καθυστέρησης που εισάγει το μοντέλο μέχρι να υπολογίσει μια πιθανότητα χρησιμοποιήθηκε αντί για την κλασσική μέθοδο *.predict()* του *Keras* απί μια διαφορετική προσέγγιση. Ειδικότερα ο τελικός 3D πίνακας με τα κανονικοποιημένα δεδομένα του παραθύρου, 3D διότι τέτοια διάσταση “απαιτεί” ένα *LSTM*, μετατράπηκε σε τανυστή της *TensorFlow*, οι τανυστές είναι τα βασικά αντικείμενα που χρησιμοποιεί αυτή η βιβλιοθήκη για την υλοποίηση όλων των υπολογισμών και ενεργειών. Στην συνέχεια αυτός ο τανυστής “περάστηκε” στο μοντέλο για την πρόβλεψη μέσω απευθείας κλήσης του αντικειμένου που το περιγράφει, *my_model(input_tensor, training = False)*, πρακτικά έτσι το αντικείμενο συμπεριφέρεται ως συνάρτηση ενώ με αυτήν την κλήση επί της ουσίας χρησιμοποιείται η μέθοδος *__call__* του αντικειμένου. Αυτή η προσέγγιση προτιμάται όταν πρέπει η πρόβλεψη να πραγματοποιηθεί σε πραγματικές-“ζωντανές” συνθήκες, όπου το μοντέλο τροφοδοτείται συνήθως με ένα δείγμα την φορά, διότι η μέθοδος *.predict()* είναι σχεδιασμένη για την διαχείριση εκατοντάδων/χιλιάδων δειγμάτων χωρισμένων σε *batches* με αποτέλεσμα να εκτελεί ενέργειες που είναι αχρείαστες όταν διαθέτουμε μόνο ένα δείγμα. Σε αντίθεση η μέθοδος που περιγράφηκε “παρακάμπτει” όλες αυτές τις ενέργειες και κατά συνέπεια είναι πολύ πιο γρήγορη υπολογιστικά. Συνεχίζοντας αφού “απομονωνόταν” η πιθανότητα από τον τανυστή που επιστρέφει η προαναφερθείσα μέθοδος ακολουθούσε η εξής διαδικασία. Αυξανόταν ένας μετρητής κατά 1 αν αυτή η πιθανότητα ήταν μικρότερη ή ίση από μια τιμή, συγκεκριμένα το 0.3 ενώ το όριο των πιθανοτήτων προέκυψε σύμφωνα με το κεφάλαιο 5.5 0.3753, αλλιώς

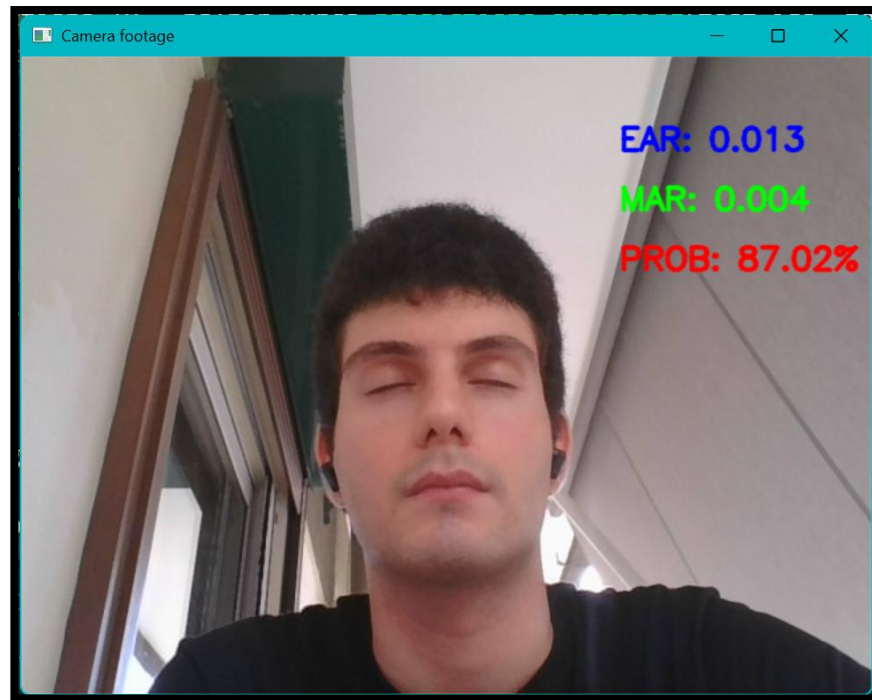
ο μετρητής μηδενιζόταν. Αν τώρα αυτός ο μετρητής γινόταν μεγαλύτερος ή ίσος από την τιμή 3, επί της ουσίας αν το μοντέλο είχε εξάγει τουλάχιστον 3 συνεχόμενες αρκετά χαμηλές πιθανότητες, κάτι το οποίο μεταφράζεται σε τουλάχιστον 1sec πραγματικού χρόνου στα 6fps, θεωρώντας με αρκετή σιγουριά ότι το άτομο μπροστά από την κάμερα είναι ξύπνιο, αφαιρούνταν όλα τα στοιχεία από την deque των πιθανοτήτων και σε κάθε θέση αυτής εισερχόταν η τελευταία χαμηλή πιθανότητα που παρήγαγε το μοντέλο. Σε κάθε άλλη περίπτωση η πιθανότητα απλά εισερχόταν στα δεξιά της deque των πιθανοτήτων. Αυτό έγινε σε μια προσπάθεια μείωσης της χρονικής καθυστέρησης, ορισμένες φορές, στην αναγνώριση της εγρήγορσης του ατόμου έπειτα από επεισόδιο υπνηλίας, με βάση τις “ωμές” πιθανότητες. Συγκεκριμένα από την στιγμή που η κάθε νέα πιθανότητα που εξάγει το μοντέλο “εξομαλύνεται”, πριν χρησιμοποιηθεί για την αξιολόγηση της κατάστασης του ατόμου, στο μέσο ενός παραθύρου μεγέθους 13 τιμών, αναπόφευκτα εισάγεται καθυστέρηση στο σύστημα σε “ζωντανό” περιβάλλον. Ειδικότερα αν ένα παράθυρο είναι γεμάτο με χαμηλές πιθανότητες, το άτομο βρίσκεται σε εγρήγορση, από την στιγμή που θα αρχίσει π.χ. να κοιμάται για να πάρει το μέσο (median) για πρώτη φορά τιμή πάνω από το όριο (0.3753) πρέπει στο παράθυρο των πιθανοτήτων να εισαχθούν 7 συνεχόμενες πιθανότητες πάνω από το όριο. Παρόμοια είναι η διαδικασία και για το αντίθετο “σενάριο”, δηλαδή από κατάσταση υπνηλίας το άτομο ανέκτησε την εγρήγορση του. Αυτός ο απαιτούμενος αριθμός πιθανοτήτων με βήμα 2 καρέ ανάμεσα τους και ρυθμό επεξεργασίας δεδομένων στα 6fps μεταφράζεται σε περίπου 2.3sec καθυστέρησης μέχρι το μέσο του παραθύρου να αποτυπώσει την αλλαγή στην κατάσταση του οδηγού. Σε ότι αφορά τώρα την εξομάλυνση των πιθανοτήτων σημαντικό είναι ακόμα να σημειωθεί ότι το πρώτο μέσο αποτιμήθηκε αφού η deque των πιθανοτήτων είχε φτάσει το μέγιστο μέγεθος της (13), από εκεί και ύστερα το μέσο όπως αναφέρθηκε υπολογιζόταν για κάθε νέα πιθανότητα που “έμπαινε” στην δομή, κατά συνέπεια η πρώτη “εξομαλυσμένη” πιθανότητα υπολογίστηκε περίπου 4sec έπειτα από την λήξη της φάσης συλλογής δεδομένων, που περιγράφηκε παραπάνω, ενώ αποθηκευόταν και σε μια καθολική μεταβλητή για την μετέπειτα χρήση της. Κλείνοντας την περιγραφή του live demo της εργασίας οι καθολικές μεταβλητές στις οποίες αποθηκεύτηκαν η “εξομαλυσμένη” πιθανότητα, το μέσο EAR και η μοναδική τιμή του MAR, του τρέχοντος καρέ που επεξεργάστηκε, χρησιμοποιήθηκαν από τον βασικό βρόγχο της κάμερας ώστε να τυπωθούν οι τιμές τους στο συγκεκριμένο καρέ. Συγκεκριμένα μέσα στον βρόγχο της κάμερας με την βοήθεια της βιβλιοθήκης OpenCV οι τιμές αυτές εγγράφονταν σε κάθε καρέ που διαβαζόταν, παρότι οι καθολικές μεταβλητές που τις περιείχαν “ανανεώνονταν” με ρυθμό 6fps, ενώ κάθε καρέ εμφανιζόταν στο ίδιο παράθυρο της OpenCV ώστε να φαίνεται μια ροή βίντεο όπου τόσο τα καρέ όσο και οι τιμές πάνω σε αυτά ανανεώνονταν με ρυθμό 30fps καθιστώντας το αποτέλεσμα ομαλό στα μάτια του χρήστη. Σε αυτό το σημείο να διευκρινιστεί η εξής λεπτομέρεια, για τα πρώτα 10 sec πραγματικού χρόνου (60 επεξεργασμένα καρέ στα 6fps) του βρόγχου της κάμερας στα καρέ εμφανιζόταν το EAR, το MAR και η πρόοδος (σε ποσοστό τοις εκατό) της διαδικασίας της συλλογής δεδομένων για την κανονικοποίηση που αναφέρθηκε, δεν εμφανιζόταν δηλαδή η πιθανότητα από την στιγμή που δεν γινόταν κάποια πρόβλεψη. Η πιθανότητα μαζί με τους δύο μαθηματικούς λόγους τυπωνόταν στα καρέ από τα 10sec και μετά. Στην συνέχεια δίνονται εικόνες από το “περιβάλλον” του live demo:



Εικόνα 32: Φάση Συλλογής Δεδομένων



Εικόνα 33: Κατάσταση Εγρήγορης



Εικόνα 34: Κατάσταση Υπνηλίας

6. Συμπεράσματα και Μελλοντικές Προεκτάσεις

6.1: Αναφορά συμπερασμάτων και προεκτάσεων

Στην παρούσα εργασία προτάθηκε ένα σύστημα αναγνώρισης υπνηλίας βασισμένο σε βασικά σημεία-ορόσημα του προσώπου, στους μαθηματικούς-γεωμετρικούς δείκτες EAR και MAR και στην χρήση βαθιάς μάθησης. Όπως φάνηκε και από τις μετρικές που παρουσίασε το τελικό σύστημα κατά την διάρκεια της αξιολόγησης του αποτελεί μια αξιόλογη πρόταση στο πρόβλημα της αναγνώρισης της υπνηλίας, καταφέροντας ιδιαίτερα ικανοποιητικά να αναγνωρίσει την κατάσταση του οδηγού. Ακόμα το ότι το τεχνητό νευρωνικό δίκτυο που αναλαμβάνει την διάκριση της κατάστασης του οδηγού αποτελείται από μόλις γύρω στις 5.5 χιλιάδες παραμέτρους, σε συνδυασμό με το ότι τα μοντέλα της MediaPipe που χρησιμοποιούνται για την εύρεση των σημείων του προσώπου είναι παρομοίως ιδιαίτερα αποδοτικά υπολογιστικά, το καθιστά μια πρόταση που μπορεί να εκτελεστεί ακόμα και σε συσκευές με περιορισμένους υπολογιστικούς πόρους, όπως είναι το σύστημα ενός αυτοκινήτου. Από την άλλη όπως φανερώθηκε και με την βοήθεια του live demo που κατασκευάστηκε και περιγράφηκε ένας περιορισμός του συστήματος είναι σαφώς τα σκούρα γυαλιά ηλίου, σε αντίθεση με τα κλασσικά διάφανα γυαλιά οράσεως που δεν αποτελούν “εμπόδιο” για αυτό. Όταν ο οδηγός φοράει τέτοιου είδους γυαλιά αποκρύπτονται πλήρως τα μάτια του με αποτέλεσμα η αναγνώριση της υπνηλίας να βασίζεται μόνο στον δείκτη MAR και αναπόφευκτα η απόδοση του μοντέλου να “πέφτει” σε μεγάλο βαθμό. Παρομοίως η απόδοση του πέφτει όταν γενικά το στόμα ή/και τα μάτια του οδηγού καλύπτονται ή αποκρύπτονται από την κάμερα με οποιονδήποτε τρόπο, μέσα σε αυτούς συγκαταλέγεται και ο πολύ χαμηλός φωτισμός (συνθήκες πλήρους σκοταδιού), σε αντίθεση με τον μέτριο φωτισμό που δεν αποτελεί πρόβλημα. Επιπρόσθετα όχι αδιάφορο μειονέκτημα του συστήματος αποτελεί η καθυστέρηση λόγω της διαδικασίας της “εξομάλυνσης”, που πραγματοποιείται, των πιθανοτήτων που εξάγει κατά την εκτέλεση του σε “ζωντανό” περιβάλλον. Συγκεκριμένα γίνεται αναφορά στην καθυστέρηση, που περιγράφηκε, στο να αποτιμήσει το μέσο των πιθανοτήτων την αλλαγή από εγρήγορη σε υπνηλία και το αντίστροφο. Παρότι οι μετρικές του μοντέλου αν δεν πραγματοποιηθεί η διαδικασία της εξομάλυνσης δεν απέχουν πολύ σε σχέση με αυτές που παρουσιάστηκαν είναι μια διαδικασία σαφώς ωφέλιμη καθώς μείωσε τον αριθμό των λανθασμένων προβλέψεων που έχουν ιδιαίτερη βαρύτητα σε μια τέτοια εφαρμογή και απαιτείται να είναι όσο το δυνατόν λιγότερες. Τώρα λύση όταν ο οδηγός θα βρίσκεται σε πολύ χαμηλό φωτισμό θα μπορούσε να αποτελέσει η χρήση του προτεινόμενου συστήματος σε συνδυασμό με μια υπέρυθρη κάμερα, ώστε το πρόσωπο του και κατά συνέπεια τα χαρακτηριστικά του να μπορούν να είναι εμφανή ακόμα και στο σκοτάδι. Σε ότι αφορά τους υπόλοιπους περιορισμούς που αναφέρθηκαν, η καθυστέρηση της “εξομάλυνσης” θα μπορούσε θεωρητικά να εξαφανιστεί με την περαιτέρω αύξηση της απόδοσης του μοντέλου χωρίς αυτήν, προσθέτοντας επιπλέον χαρακτηριστικά πέραν των EAR και MAR, παράγωγα αυτών (συχνότητα βλεφαρισμού, χασμουρητού κ.α.), καθιστώντας την ανάγκη για “εξομάλυνση” πιθανόν περιττή. Ενώ σε ότι αφορά το ζήτημα των περιπτώσεων κατά τις οποίες αποκρύπτονται τα μάτια

ή/και το στόμα, με αποτέλεσμα να μην μπορούν να υπολογιστούν συμπεριφορικά χαρακτηριστικά και να μην μπορεί να πραγματοποιηθεί πρόβλεψη για την κατάσταση του οδηγού, μια λύση θα μπορούσε να είναι η προσθήκη και βιολογικών χαρακτηριστικών, όπως είναι ο καρδιακός ρυθμός, με την εγκατάσταση αισθητήρων στο σώμα του οδηγού, ή/και χαρακτηριστικών που προκύπτουν από τα δεδομένα του αυτοκινήτου και τα μοτίβα της κίνησης του, όπως είναι η απόκλιση του οχήματος από την λωρίδα κυκλοφορίας. Τέλος, μια ακόμα προέκταση που θα μπορούσε να πραγματοποιηθεί, έπειτα από την υλοποίηση των προαναφερθέντων, αποτελεί η προσθήκη έγκυρων, ηχητικά σωστών και χρονικά εναρμονισμένων ηχητικών-φωνητικών προειδοποιήσεων για την αποτελεσματική αφύπνιση του οδηγού, στο demo που παρουσιάστηκε.

6.2: Αποθετήριο κώδικα στο GitHub

Στο *link* που ακολουθεί βρίσκεται το αποθετήριο στο *GitHub* το οποίο περιέχει όλους τους κώδικες στην γλώσσα προγραμματισμού *Python* που υλοποιήθηκαν για τους σκοπούς της εργασίας.

[Αποθετήριο Προγραμμάτων Εργασίας](#)

7.ΒΙΒΛΙΟΓΡΑΦΙΑ

[1]

Somnolence/Drowsiness → Wikipedia (Ιστοσελίδα)

Link:

<https://en.wikipedia.org/wiki/Somnolence>

[2]

Τροχαία: Η κόπωση των οδηγών και η υπνηλία από τις βασικές αιτίες των ατυχημάτων → Protothema (Ηλεκτρονικό Άρθρο, 2022)

Link:

<https://www.protothema.gr/greece/article/1299404/trohaia-i-koposi-ton-odigon-kai-i-upnilia-apo-tis-vasikes-aities-ton-atuhimaton/>

[3]

Τεχνητή νοημοσύνη → Βικιπαίδεια (Ιστοσελίδα)

Link:

<https://el.wikipedia.org/wiki/%CE%A4%CE%B5%CF%87%CE%BD%CE%B7%CF%84%CE%AE%CE%BD%CE%BF%CE%B7%CE%BC%CE%BF%CF%83%CF%8D%CE%BD%CE%B7>

[4]

ML: Common Loss Functions → GeeksforGeeks (Ιστοσελίδα, 2025)

Link:

<https://www.geeksforgeeks.org/machine-learning/ml-common-loss-functions/>

[5]

A Brief Overview of Cross Entropy Loss → Medium (Ιστοσελίδα, 2024)

Link:

<https://medium.com/@chris.p.hughes10/a-brief-overview-of-cross-entropy-loss-523aa56b75d5>

[6]

Types of Optimizers in Deep Learning: A Comprehensive Guide → Medium (Ιστοσελίδα, 2025)

Link:

<https://medium.com/@sushmita2310/types-of-optimizers-in-deep-learning-a-comprehensive-guide-af258d6acf66>

[7]

Introduction to Long Short Term Memory → GeeksforGeeks (Ιστοσελίδα, 2026)

Link:

<https://www.geeksforgeeks.org/deep-learning/deep-learning-introduction-to-long-short-term-memory/>

[8]

TensorFlow → Wikipedia (Ιστοσελίδα)

Link:

<https://en.wikipedia.org/wiki/TensorFlow>

[9]

Keras: The high-level API for TensorFlow → TensorFlow (Ιστοσελίδα, 2023)

Link:

<https://www.tensorflow.org/guide/keras#models>

[10]

M. Ramzan, H. U. Khan, S. M. Awan, A. Ismail, M. Ilyas and A. Mahmood, "A Survey on State-of-the-Art Drowsiness Detection Techniques," in *IEEE Access*, vol. 7, pp. 61904-61919, 2019

[11]

Kazemi, Vahid & Sullivan, Josephine. (2014). One Millisecond Face Alignment with an Ensemble of Regression Trees.

[12]

Lugaresi, Camillo et al. "MediaPipe: A Framework for Building Perception Pipelines." *ArXiv abs/1906.08172* (2019): n. pag

[13]

Kartynnik, Yury et al. "Real-time Facial Surface Geometry from Monocular Video on Mobile GPUs." *ArXiv abs/1907.06724* (2019): n. pag.

[14]

Dinges, David F. and Randolph C. Grace. "Perclos: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance." (1998).

[15]

Soukupová, Tereza and Jan Cech. "Real-Time Eye Blink Detection using Facial Landmarks." (2016).

[16]

Bamidele, Ameen Aliu et al. "Non-intrusive Driver Drowsiness Detection based on Face and Eye Tracking." *International Journal of Advanced Computer Science and Applications* (2019): n. pag.

[17]

Majeed F, Shafique U, Safran M, Alfarhood S, Ashraf I. Detection of Drowsiness among Drivers Using Novel Deep Convolutional Neural Network Model. *Sensors*. 2023; 23(21):8741.

[18]

Hu He, Xiaoyong Zhang, Fu Jiang, Chenglong Wang, Yingze Yang, Weirong Liu, Jun Peng, A Real-time Driver Fatigue Detection Method Based on Two-Stage Convolutional Neural Network, *IFAC-PapersOnLine*, Volume 53, Issue 2, 2020, Pages 15374-15379.

[19]

Chen, Long, Xin, Guojiang, Liu, Yuling, Huang, Junwei, Driver Fatigue Detection Based on Facial Key Points and LSTM, *Security and Communication Networks*, 2021, 5383573, 9 pages, 2021.

[20]

Abtahi, Shabnam et al. "YawDD: a yawning detection dataset." *ACM SIGMM Conference on Multimedia Systems* (2014).

[21]

Ghoddosian, Reza et al. "A Realistic Dataset and Baseline Temporal Model for Early Drowsiness Detection." 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2019): 178-187.

[22]

Weng, Ching-Hua et al. "Driver Drowsiness Detection via a Hierarchical Temporal Deep Belief Network." ACCV Workshops (2016).

[23]

Face landmark detection guide → Google For Developers (Ιστοσελίδα, 2026)

Link:

https://developers.google.com/edge/mediapipe/solutions/vision/face_landmarker

[24]

Driver Drowsiness Dataset → Kaggle (Ιστοσελίδα-Πλατφόρμα)

Link:

<https://www.kaggle.com/datasets/azrilfahmiardi/driver-drowsiness/data>