



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορικής»
ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ

Τίτλος Διατριβής	Αναγνώριση Συναισθημάτων μέσω Εικόνων με την χρήση Συνελικτικών Νευρωνικών Δικτύων Emotion Recognition through Images using Convolutional Neural Networks
Ονοματεπώνυμο Φοιτητή	Βισβάρδης Αναστάσιος
Πατρώνυμο	Διονύσιος
Αριθμός Μητρώου	ΜΠΠΛ2209
Επιβλέπων Καθηγητής	Σωτηρόπουλος Αναστάσιος Αναπληρωτής Καθηγητής

Ημερομηνία Παράδοσης May 2026 / Μάιος 2026

Τριμελής Εξεταστική Επιτροπή

Διονύσιος Σωτηρόπουλος
Αναπληρωτής Καθηγητής

Γεώργιος Τσιχριντζής
Καθηγητής

Ευάγγελος Σακκόπουλος
Αναπληρωτής Καθηγητής

Περιεχόμενα

Περίληψη.....	
1 Εισαγωγή.....	
1.1 Αντικείμενο-Στόχοι της Εργασίας.....	
1.2 Μεθοδολογία.....	
1.3 Διατύπωση-Σημαντικότητα Προβλήματος.....	
1.4 Τι είναι Συναισθημα.....	
1.5 Εξέλιξη της έννοιας ‘Συναισθημα’.....	
1.6 Κατηγορίες.....	
1.7 Cognitive Approach.....	
2 Affective Computing.....	
2.1 Τι είναι το Affective Computing.....	
2.2 Emotion Models.....	
2.3 Affective Computing and Sentiment Analysis.....	
2.4 Sentiment Analysis and Neural Networks.....	
3 Συνελικτικά Νευρωνικά Δίκτυα.....	
3.1 Τι είναι τα Συνελικτικά Δίκτυα.....	
3.2 Διαφορές FC-NN's με CNN's	
3.3 Χαρακτηριστικά CNN.....	
3.3.1 Convolutional Layers.....	
3.3.2 Max Pooling.....	
3.3.3 Activation Functions.....	
4 Dataset-Dataset Preprocessing.....	
4.1 Διαδεδομένα Datasets για emotion recognition και τα χαρακτηριστικά τους.....	

4.2 Επεξεργασία Δεδομένων.....	
4.3 Data Augmentation.....	
4.4 Class Imbalance.....	
5 Experimental Design-Experimental Results.....	
5.1 Dataset.....	
5.1.1 Dataset πριν την Επεξεργασία....	
5.1.2 Επεξεργασία Dataset	
5.2 Σχεδιασμός Συνελικτικού Δικτύου.....	
5.3 Αποτελέσματα.....	
6 Συμπεράσματα και Μελλοντική Έρευνα.....	
6.1 Συμπεράσματα.....	
6.2 Μελλοντικές Προεκτάσεις.....	
6.3 Τελικές Παρατηρήσεις.....	
6.4 Συγκεντρωτικός Πίνακας.....	
7 Βιβλιογραφία	

Περίληψη

Η παρούσα εργασία εστιάζει στον τρόπο με τον οποίο τα νευρωνικά δίκτυα και ποιο συγκεκριμένα τα Συνελικτικά νευρωνικά δίκτυα αναγνωρίζουν τα διάφορα συναισθήματα μέσω ενός συγκεκριμένου Dataset εικόνων. Στο πρώτο κεφάλαιο γίνεται μια εισαγωγή στο αντικείμενο της εργασίας, καθώς επίσης διατυπώνεται το πρόβλημα και γίνεται μια ανάλυση της σημαντικότητάς του (Ανάλυση της έννοιας

‘Συναίσθημα’). Στο δεύτερο κεφάλαιο αναλύεται η έννοια του ‘Affective Computing’, η σχέση της μηχανικής μάθησης με την έννοια του συναισθήματος καθώς και η συσχέτιση του ‘Affective Computing’ με τα νευρωνικά δίκτυα. Στο Κεφάλαιο 3, εισάγεται ο όρος των ‘Συνελικτικών Νευρωνικών Δικτύων’. Αναλύονται οι διαφορές των ‘Fully Connected Neural Networks(FNN’s)’ σε σχέση με τα ‘Convolutional Neural Networks(CNN’s)’, καθώς επίσης και τα κύρια χαρακτηριστικά των CNN’s. Στο

κεφάλαιο 4 εξετάζονται τα πιο βασικά και διαδεδομένα 'Datasets Εικόνων', η διαδικασία της επεξεργασίας των εικόνων, η έννοια του 'Data Augmentation' καθώς και τους διαφορετικούς τρόπους που αντιμετωπίζουμε το 'Class Imbalance' στο dataset μας. Στο κεφάλαιο 5 εξηγείται το Experimental Design του Project που έχω φτιάξει και τα Experimental Results του. Στο τελευταίο κεφάλαιο, το 7 εξάγω τα συμπεράσματα, προτείνω μελλοντικές προεκτάσεις-βελτιώσεις και παραθέτω τις τελικές παρατηρήσεις. Τέλος παρατίθεται η Βιβλιογραφία.

Abstract

The present work focuses on the way neural networks, specifically Convolutional Neural Networks, recognize various emotions through a specific dataset of

images. The first chapter provides an introduction to the subject of the work, while also formulating the problem and analyzing its significance (Analysis of the concept of 'Emotion'). In the second chapter, the concept of 'Affective Computing' is analyzed, as well as the relationship between machine learning and the concept of emotion, and the correlation between 'Affective Computing' and neural networks. In Chapter 3, the term 'Convolutional Neural Networks' is introduced. The differences between 'Fully Connected Neural Networks (FNNs)' and 'Convolutional Neural Networks (CNNs)' are analyzed, why Convolutional Neural Networks are the most suitable when our dataset consists of images, as well as the main characteristics of CNNs. In chapter 4, the most basic and widespread 'Image Datasets' are examined, the process of image processing, the concept of 'Data Augmentation', as well as the different ways we address 'Class Imbalance' in our dataset. In chapter 5, the Experimental Design of the Project I have created and the Experimental Results of this Project are explained. In the final chapter, chapter 7, I draw conclusions, suggest future extensions/improvements, and present final observations. Finally, the Bibliography is appended.

Αναγνώριση Συναισθημάτων μέσω Εικόνων με την χρήση Συνελικτικών Νευρωνικών Δικτύων.

Βισβάρδης Αναστάσιος-ΜΠΠΛ2209

1. Εισαγωγή

1.1 Αντικείμενο-Στόχοι της Εργασίας

Το αντικείμενο της παρούσας μεταπτυχιακής διατριβής αποτελεί μιας πλήρους και εμπειριστατωμένης ανασκόπησης του τρόπου με τον οποίο συνδέονται οι συναισθηματικές αποκρίσεις με την μηχανική μάθηση και ποιο συγκεκριμένα με τα Νευρωνικά Δίκτυα. Η παρούσα εργασία χωρίζεται σε δύο επίπεδα: **(α)** Ένα θεωρητικό, στο οποίο γίνεται μια ανάλυση των διαφόρων συναισθηματικών θεωριών, την σύνδεσή τους με την μηχανική μάθηση και την ανάλυση των Συνελικτικών νευρωνικών δικτύων **(β)** Ένα Συνελικτικό Δίκτυο το οποίο προσπαθεί τα μετουσιώσει την θεωρητική ανάλυση στην πράξη, να εξάγει συμπεράσματα σε σχέση με τον βαθμό σύνδεσης των δύο επιστημονικών πεδίων(Επιστήμη των συναισθημάτων και Μηχανική Μάθηση),να εξετάσει διεξοδικά τα αποτελέσματα του project, να εξάγει συμπεράσματα από τα αποτελέσματα που θα προκύψουν και να προτείνει μελλοντικές προεκτάσεις.

Σύμφωνα με αυτά που αναφέρθηκαν, στόχοι της παρούσα διπλωματικής είναι:

- i) Η ακριβής και αποδοτική παρουσίαση του θεωρητικού υποβάθρου και η σύνδεση του με το πρακτικό.
- ii) Η Εξαγωγή συμπερασμάτων και η εμπειριστατωμένη ανάλυσή τους.
- iii) Προτάσεις για την βελτίωση του είδη υπάρχοντα κώδικα.

1.2 Μεθοδολογία

Η μεθοδολογία που χρησιμοποιήθηκε στο πρακτικό επίπεδο μπορεί να χωριστεί σε τρία κομμάτια: 1) Την εύρεση και την δοκιμή διαφόρων Datasets με σκοπό να βρεθεί το κατάλληλο, που να τηρεί τις προδιαγραφές για την εξαγωγή ασφαλών συμπερασμάτων, με τελική επιλογή να είναι το RAF-DB DATASET. 2) Την επεξεργασία του Dataset με την χρήση μεθόδων όπως: α) Ενοποίηση ομοειδών συναισθηματικών εκφράσεων σε έναν φάκελο με σκοπό την αντιμετώπιση της ανισορροπίας των κλάσεων β) Την εφαρμογή της μεθόδου του Oversampling, ώστε ο αριθμός των δειγμάτων σε κάθε φάκελο να μην αποκλίνει πολύ από τον

φάκελο με τα περισσότερα δείγματα. γ) Την εφαρμογή της μεθόδου του Class Weight στις εικόνες, μειώνοντας έτσι κι' άλλο τις αποκλίσεις, προσδίδοντας διαφορετικά 'βάρη' σε κάθε κατηγορία. 2) Μετά την επεξεργασία του Dataset το επόμενο βήμα ήταν η δημιουργία του Νευρωνικού Δικτύου. Τα Convolutional Layers και το Max Pooling επιλέχθηκαν προσεκτικά, ώστε να προσαρμόζονται στα δεδομένα μου. Χρησιμοποιήθηκαν 'Callbacks' όπως τα 'EarlyStopping' και 'ReduceLROnPlateau' με σκοπό τον έλεγχο του Overfitting. Τέλος, εφαρμόστηκε η μέθοδος του 'Finetuning', με την επανεκπαίδευση του μοντέλου με σκοπό να μάθει περισσότερες λεπτομέρειες σε σχέση με τα δεδομένα που του δόθηκαν.

1.3 Διατύπωση-Σημαντικότητα του Προβλήματος

Διατύπωση του Προβλήματος

Το πρόβλημα που μελετάτε στην συγκεκριμένη μεταπτυχιακή διατριβή αποτελεί έναν πολύπλοκο γρίφο από την αρχή, σχεδόν της ανθρώπινης ύπαρξης. Η επιστήμη των συναισθημάτων αποτελεί ένα γνωστικό πεδίο το οποίο συνδυάζει διάφορα άλλα επιστημονικά πεδία, όπως ψυχολογία, κοινωνιολογία, επιστήμη των υπολογιστών κλπ, ώστε να μπορεί να εξεταστεί στην ολότητά του. Τα θεωρητικά μοντέλα που έχουν αναπτυχθεί είναι πολλές φορές αντικρουόμενα και δύσκολα μπορούν να 'μεταφραστούν' σε υπολογιστικά συστήματα. Μ' αυτό σαν δεδομένο, την διεπιστημονικότητα του συγκεκριμένου πεδίου, την υποκειμενικότητα που υπάρχει στον τρόπο με τον οποίο εκλαμβάνονται τα συναισθήματα ανάλογα το πολιτισμικό πλαίσιο καθώς και την πολυπλοκότητα των διαφόρων συναισθηματικών εκφράσεων, η δημιουργία ενός υπολογιστικού συστήματος που θα μπορεί να προβλέπει σωστά τις διάφορες συναισθηματικές εκφράσεις αποτελεί πρόκληση για την μηχανική μάθηση.

Σημαντικότητα του Προβλήματος

Παρόλες τις προκλήσεις που εγείρονται εξετάζοντας το συγκεκριμένο αντικείμενο, η μελέτη του αποτελεί καίρια, καθώς οι εφαρμογές του εντοπίζονται σε διάφορα επίπεδα της ανθρώπινης ύπαρξης.

(α) **Ψυχική υγεία:** Η 'Real-Time' πρόβλεψη των συναισθηματικών αντιδράσεων ενός ασθενή μπορεί να συμβάλει θετικά στον έργο των επιστημόνων της ψυχικής υγείας, δίνοντάς τους ένα επιπλέον εργαλείο.

(β) **Εκπαίδευση:** Μοντέλα αναγνώρισης συναισθημάτων μπορούν να χρησιμοποιηθούν κατά την διάρκεια της εκπαιδευτικής διαδικασίας, δίνοντας έτσι την

δυνατότητα στο εκπαιδευτικό προσωπικό να εξάγει καλύτερα συμπεράσματα για τον τρόπο με τον οποίο αντιδρούν οι μαθητές στα διάφορα ερεθίσματα που δέχονται.

(γ) **Μάρκετινγκ:** Δίνεται η δυνατότητα σε επιχειρήσεις να παρακολουθούν την συναισθηματική αντίδραση των καταναλωτών στα προϊόντα και τις υπηρεσίες.

(δ) **Αλληλεπίδραση Ανθρώπου-Υπολογιστή:** Η διερεύνηση του συγκεκριμένου αντικειμένου συμβάλει καλύτερα στον τρόπο με τον οποίο συσχετίζεται ο άνθρωπος με τον υπολογιστή. Η ανάλυση των συναισθημάτων για παράδειγμα κατά την διάρκεια της

περιήγησης ενός χρήστη στο διαδίκτυο, μπορεί να βοηθήσει τον πάροχο να προσφέρει εξατομικευμένο περιεχόμενο στον χρήστη και τον τελευταίο να επιλέξει το καταλληλότερο γι' αυτόν.

1.4 Τι είναι το Συναίσθημα

Είναι κοινός διαδεδομένο ότι αυτό που ξεχωρίζει τους ανθρώπους από τα υπόλοιπα έμβια όντα σ' αυτόν τον πλανήτη είναι η δυνατότητα του να σκέφτονται ορθολογικά, να μπορούν συνήθως, να ξεετάζουν όλες τις πιθανότητες που υπάρχουν και να παίρνουν την πιο λογική επιλογή. Ο συγκεκριμένος ισχυρισμός παρόλο που εμπεριέχει μια δόση αλήθειας, στην πραγματικότητα αποτελεί μόνο την 'άκρη του παγόβουνου', του τρόπου με τον οποίο οι άνθρωποι παίρνουν αποφάσεις, καθώς δεν βάζει στην εξίσωση το συναίσθημα.

Στην πραγματικότητα οι περισσότερες αποφάσεις δεν ελέγχονται συνειδητά από τον άνθρωπο. Η 'κινητήριος δύναμη', που υποσυνείδητα καθοδηγεί τις επιλογές μας χωρίς πολλές φορές να το καταλαβαίνουμε είναι το 'Συναίσθημα'. Ετυμολογικά, η λέξη συναίσθημα(Emotion) χωρίζεται σε δύο συνθετικά στο 'ε', που υποδηλώνει την κατεύθυνση και στο 'movere', που σημαίνει κίνηση. Αυτά τα δύο μαζί ουσιαστικά περιγράφουν αυτό που το συναίσθημα κάνει, μας καθοδηγεί σε κίνηση ως προς κάποιο εξωτερικό ή και εσωτερικό ερέθισμα[1].

Χωρίς τα συναισθήματα ή ένστικτο όπως πολλές φορές αναφέρονται οι διάφορες συναισθηματικές αντιδράσεις, για να περιγράφουν ποιο απλά το ανθρώπινο είδος δεν θα μπορούσε να εξελιχθεί σ' αυτό που είναι τώρα. Έχει αποδειχθεί ότι στα εξωτερικά ερεθίσματα που δεχόμαστε πρώτα αντιδρούμε συναισθηματικά και μετά επεξεργάζεται την κατάσταση ο λογικός εγκέφαλος[2].

Σύμφωνα με τον Newell, τα συναισθήματα εκδηλώνονται σε τέσσερα επίπεδα[2]:

- 1) **Βιολογικό:** Για παράδειγμα αν δούμε ένα λιοντάρι, ενεργοποιείται το συναίσθημα του φόβου. Υπάρχει άμεση ετοιμότητα του συστήματος 'Fight-or-Flight', με σκοπό να αντιμετωπιστεί ο κίνδυνος. Η καρδιά χτυπάει πιο δυνατά, η αναπνοή μας είναι πιο γρήγορη, η θερμοκρασία του σώματος αυξάνεται, οι κόρες των ματιών διαστέλλονται καθιστώντας έτοιμο το σώμα στον αναμενόμενο κίνδυνο.
- 2) **Γνωσιακό:** Τα συναισθήματα διαμορφώνουν σε πολύ μεγάλο βαθμό τον τρόπο με τον οποίο σκεφτόμαστε. Για παράδειγμα, όταν θυμώνουμε τείνουμε να ανασύρουμε αναμνήσεις που ενισχύουν το συναίσθημα που νιώθουμε εκείνη την στιγμή. Επίσης ο χρόνος δράσης σ' αυτή την περίπτωση είναι πολύ μικρός σε σχέση με μια 'νηφάλια' στιγμή.
- 3) **Λογικό:** Τα συναισθήματα έρχονται σε δεύτερη μοίρα, παρόλο που στην πραγματικότητα είναι η κινητήριος δράση. Για παράδειγμα, η απόφαση για το αν θα παραιτηθούμε ή όχι από την εργασία μας,

περνάει από πολλά φίλτρα επεξεργασίας, αλλά την τελική απόφαση την έχει η λογική.

- 4) **Κοινωνικό:** Τα συναισθήματα, πέρα από μέσο επιβίωσης, αποτελούν έναν από τους κύριους τρόπους επικοινωνίας στον σύγχρονο πολιτισμό. Για παράδειγμα, άτομα με μεγάλη συναισθηματική νοημοσύνη(ενσυναίσθηση, συμπόνια κλπ) τείνουν να έχουν ποιο επιτυχημένες κοινωνικές και επαγγελματικές σχέσεις σε σχέση με άτομα που το επίπεδο της συναισθηματικής τους νοημοσύνης είναι χαμηλότερο.

1.5 Εξέλιξη της Έννοιας 'Συναίσθημα'

Ανέκαθεν οι άνθρωποι, από όταν δημιούργησαν πολιτισμό προσπαθούσαν να εξηγήσουν το τι είναι το συναίσθημα. Από την αρχαιότητα μέχρι το σήμερα η αποκωδικοποίηση των τρόπων που δρούμε, σκεφτόμαστε, επικοινωνούμε, μαθαίνουμε και στην τελική ανάλυση υπάρχουμε, αποτελεί καίριο πρόβλημα της ανθρωπίνης ύπαρξης.

Διαχρονικά έχουν αναπτυχθεί διάφορες θεωρίες αναλόγως των συνθηκών που επικρατούσαν τις ιστορικές περιόδους που διατυπώθηκαν. Ποιο συγκεκριμένα[3]:

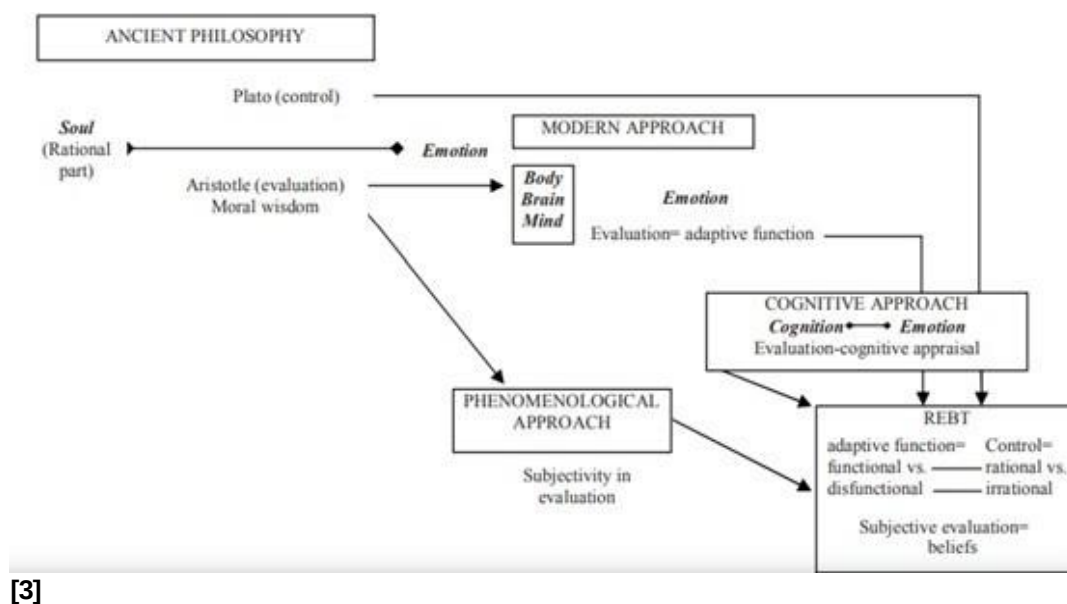
(α) **Αρχαιότητα:** Οι κύριοι φιλόσοφοι που ασχολήθηκαν με τον συναισθηματικό κόσμο στην αρχαιότητα ήταν ο Πλάτωνας και ο Αριστοτέλης. Ο Πλάτωνας θεωρούσε ότι η ψυχή είναι τριμερής(rational, emotional, appetitive) και ότι ο λόγος πρέπει να ελέγχει και να καθοδηγεί το συναίσθημα. Οπότε ο Πλάτωνας έβλεπε το συναίσθημα γενικά σαν αδυναμία και κάτι το οποίο θα έπρεπε να τιθασευτεί. Από την άλλη ο Αριστοτέλης, θεωρούσε τα συναισθήματα μέρος της ανθρωπίνης λειτουργίας εντάσσοντάς στα ως έναν από τους κύριους μηχανισμούς της ανθρωπίνης ύπαρξης.

(β) **Μοντέρνα Προσέγγιση:** Η μοντέρνα προσέγγιση άρχισε να καταρρίπτει ουσιαστικά την έννοια της 'ψυχής', που ήταν η κεντρική ιδέα για την προέλευση των συναισθημάτων στην αρχαιότητα και άρχισε να ασχολείται με το πώς δουλεύει το συναίσθημα. Ο Ντεκάρτ είναι αυτός, που πρώτος ουσιαστικά προσπαθεί να αποδώσει το συναίσθημα σε φυσιολογικούς μηχανισμούς του εγκεφάλου. Ακολουθώντας οι James-Lang και Cannon-Bard ασχολήθηκαν με το τι προηγείται, η συναισθηματική αντίδραση σ' ένα ερέθισμα ή η αντίδραση στο σώμα πρώτα. Τέλος οι Nesse&Williams ακολουθώντας την θεωρία του Δαρβίνου περί της εξέλιξης των ειδών εστίασαν την μελέτη τους στην σχέση συναισθήματος και εξωτερικού περιβάλλοντος.

(γ) Φαινομενολογική: Ακολουθώντας το ευρύτερο ρεύμα το μεταμοντερνισμού η φαινομενολογική αντίληψη 'έστρεψε' το ερώτημα του 'τι είναι συναίσθημα' ή 'πώς λειτουργεί το συναίσθημα' στο ερώτημα 'Πώς Βιώνεται'. Για τον Ζαν Πωλ Σατρ, το συναίσθημα αποτελεί στην ουσία του μια στρατηγική του εγώ με σκοπό να αντιμετωπίσει έναν παράλογο κόσμο.

(δ) Γνωστική: Με την γνωστική προσέγγιση ουσιαστικά αναδιατυπώνεται το αρχαίο δίπολο της διαφοράς ψυχής-σώματος με σύγχρονους επιστημονικούς όρους μετατοπίζοντάς το στο δίπολο γνώση-συναίσθημα. Σύμφωνα με τους Oatley και Johnson-Laird, τα συναισθήματα , μπορούν να επικοινωνηθούν με δύο τρόπους: Σε συμβολικό επίπεδο και σε συμβολικό. Τέλος ο Lazarus παρουσίασε την ιδέα της γνωστικής αξιολόγησης, δηλαδή την διαδικασία με την οποία αξιολογούμε τα συναισθήματα με βάση τους στόχους μας, αξίες μας κλπ.

(ε) Θεωρία της ορθολογικής συναισθηματικής συμπεριφοράς: Η θεωρία αυτή ουσιαστικά, στρέφει την κάμερα στο υποκείμενο. Θεωρεί ότι δεν έχει σημασία τι δημιουργεί μια συναισθηματική κατάσταση, αλλά ο τρόπος που επεξεργάζεται το άτομο το συναίσθημα, ότι οι μηχανισμοί που ενεργοποιούνται σε κάθε συναισθηματικό στάδιο είναι γνωστοί και με παρέμβαση μπορεί να αλλάξει το βίωμα του υποκειμένου.



1.6 Κατηγορίες Συναισθημάτων

Πέρα από την κατηγοριοποίηση των συναισθημάτων σε εννοιολογικό επίπεδο(Βιολογικό, Γνωσιακό, Κοινωνικό και Λογικό) και τις διαφορετικές προσεγγίσεις που ακολουθήθηκαν σε διάφορες ιστορικές περιόδους, υπάρχει και μια πάρα πολύ σημαντική κατηγοριοποίηση των συναισθημάτων, που αφορά αυτά καθ' αυτά τα συναισθήματα: Το δίπολο 'Απλό-Σύνθετο'. Το συγκεκριμένο δίπολο είναι κάτι το οποίο έχει απασχολήσει σε μεγάλο βαθμό τους επιστήμονες ου συναισθήματος.

Απαρχή της μελέτης των συναισθηματικών εκφράσεων σε εμπειρικό επίπεδο και της βάσης για την εμφάνιση του διλήματος 'Απλό-Σύνθετο', είναι το πείραμα που διεξήγαγε ο Charles Darwin χρησιμοποιώντας τις φωτογραφίες του Νευρολόγου Guillaume-Benjamin-Amand Duchenne και δημοσίευσε τα αποτελέσματα του στο έργο του, *The Expression of the Emotions in Man and Animals*. Ο Δαρβίνος στο συγκεκριμένο πείραμα χρησιμοποίησε κάποιες από τις φωτογραφίες, ρωτώντας μια ομάδα ανθρώπων να μαντέψουν το συναίσθημα που φέρνει ταραχή στον εικονιζόμενο ηλικιωμένο άνθρωπο σε κάθε εικόνα. Τα επίπεδα επιτυχής πρόβλεψης ήταν εξαιρετικά υψηλά με αποτέλεσμα να επιβεβαιωθεί η σύνδεση των συναισθημάτων με την θεωρία της εξέλιξης[4].

Πατώντας και πάνω σε αυτό το πείραμα άρχισε να γίνεται μια ευρύτερη μελέτη πάνω στα συναισθήματα(χαρά, λύπη, φόβος κλπ), καθώς και στο δίπολο 'Απλό-Σύνθετο'. Ποιο συγκεκριμένα οι κύριες θεωρήσεις πάνω σε αυτό είναι:

- 1) **Βιολογική Οπτική** : Η θεωρία αυτή ουσιαστικά εισάγει την ιδέα των βασικών συναισθημάτων. Συναισθήματα που είναι έμφυτα, από τα οποία προέρχονται όλα τα υπόλοιπα συναισθήματα. Η πιο βασική θεωρία είναι αυτή του Ekman. Ο Ekman χώριζε τα συναισθήματα σε βασικά και δευτερεύοντα. Τα βασικά είναι έξι: χαρά, λύπη, φόβος, θυμός, αηδία και έκπληξη. Τα υπόλοιπα συναισθήματα είναι συνδυασμός των 6 βασικών.
- 2) **Στοιχειώδη Προσέγγιση**: Η Στοιχειώδης προσέγγιση αποτελεί ουσιαστικά προέκταση της θεωρίας του Ekman, περί της ύπαρξης 6 βασικών συναισθημάτων. Ο Plutchik, πατώντας πάνω στην θεωρία του Ekman, εισήγαγε την ιδέα του 'μοντέλου Ανάμειξης'. Τα συναισθήματα, σε αυτή την θεωρία είναι σαν τα χρώματα. Τα σύνθετα προέρχονται από την ανάμειξη δύο ή περισσότερων βασικών. Για παράδειγμα, η ζήλια, μπορεί να είναι συνδυασμός θλίψης, θυμού και αηδίας.
- 3) **Εννοιολογικό Επίπεδο**: Σε αντίθεση με τις δύο προηγούμενες προσεγγίσεις, η συγκεκριμένη ενώ αποδέχεται ως ένα βαθμό την ύπαρξη βασικών-έμφυτων συναισθημάτων, δεν εστιάζει σε αυτό αλλά στον τρόπο με τον οποίο βιώνονται και νοηματοδοτούνται αυτά από τους ανθρώπους.

1.7 Γνωσιακή Προσέγγιση

Στην συγκεκριμένη ενότητα θα γίνει μια ποιο αναλυτική αναφορά στην μέθοδο της Γνωσιακής Προσέγγισης. Πέρα από το συναίσθημα, καθ' αυτό, σαν έννοια το τελικό project είναι ένα νευρωνικό δίκτυο το οποίο μέσω ενός Dataset, εικόνων μαθαίνει να αναγνωρίζει τα συναισθήματα. Άρα η γνωσιακή θεωρία μας είναι αρκετά κομβική στο να δούμε πώς η μάθηση συνδέεται με το συναίσθημα.

Η μάθηση και η γνώση δεν αποτελούν στην πραγματικότητα δύο πράγματα ξέχωρα από τα συναισθήματα. Ανάλογα το πως δίνεται σε λεκτικό επίπεδο στο δέκτη μια πληροφορία μπορεί να βγάλει διαφορετικά συμπεράσματα, τα οποία εξαρτώνται από το περιεχόμενο της πληροφορίας, από το πλαίσιο στο οποίο δίνεται αυτή και το συναισθηματικό 'βάρος' των λέξεων που εμπεριέχει. Για παράδειγμα, η έκφραση <<Αυτός ο σκύλος γαβγίζει>>, έχει μεγάλη διαφορά από το <<Αυτός ο σκύλος γαβγίζει επικίνδυνα>>. Η πρώτη, ανάλογα και το πλαίσιο μπορεί να εκληφθεί σαν μια ουδέτερη πληροφορία. Αντιθέτως η δεύτερη λόγο της λέξης 'επικίνδυνα' δίνει μια συναισθηματική φόρτιση στην πληροφορία. Η λέξη <<Επικίνδυνα>> σ' αυτό το πλαίσιο δεν αποτελεί απλώς μια ιδιότητα του σκύλου, αλλά ένα Σημείο Προσαρμογής(Point of Adaptation), που καλεί για δράση τον δέκτη[1].

Το Σημείο Προσαρμογής(Point of Adaptation), ουσιαστικά εξετάζει την γνώση και το συναίσθημα από την υποκειμενική σκοπιά του ατόμου. Θέτει το ερώτημα <<Σημαντικό για ποιόν και σε ποια κατάσταση>>. Τα 'Σημεία προσαρμογής' μπορούν να οριστούν και ως εκείνα τα 'φίλτρα' με τα οποία το άτομο επεξεργάζεται τις πληροφορίες που λαμβάνει από το περιβάλλον και μπορούν να χωριστούν σε δύο κατηγορίες[1]:

(α) Βασικά: Τα συγκεκριμένα 'φίλτρα' μπορούμε να τα δούμε στα περισσότερα έμβια όντα στον πλανήτη. Τα 2 ποιο χαρακτηριστικά είναι το ένστικτο της επιβίωσης και η ανάγκη της αναπαραγωγής του είδους[1].

(β) Σύνθετα: Αυτά τα 'φίλτρα' αποτελούν ξεκάθαρα ανθρώπινα κατασκευάσματα. Παραδείγματα αυτών είναι η κοινωνική θέση του ατόμου, η εκπλήρωση των προσωπικών στόχων, η οικονομική ευημερία, η ανάγκη για αλήθεια και δικαιοσύνη κλπ[1].

Τα συναισθήματα κατά κύριο λόγο δημιουργούνται από ερεθίσματα του εξωτερικού περιβάλλοντος και προκαλούν αλλαγές στην αξιολογική κρίση, στις επιλογές δράσεις, στο πως αντιδράει το σώμα και στις εκφράσεις του προσώπου[5]. Ποιο συγκεκριμένα η αλλαγές στην αξιολογική κρίση(Appraisal Theories), αποτελούν το κυριότερο εργαλείο σύνδεσης της γνώσης με το συναίσθημα. Δεν στέκονται απλώς στο πώς γεννιέται το συναίσθημα, αλλά στο γιατί για παράδειγμα μια κατάσταση δημιουργεί θλίψη, ενώ μια άλλη οργή. Υπάρχουν δύο κύριες προσεγγίσεις όσον αναφορά τις Appraisal Theories:

- 1) Συμβατότητα με τα Κύρια Συναισθήματα:** Αυτή η προσέγγιση ενσωματώνει την θεωρία του Ekman όσον αναφορά τα <<βασικά συναισθήματα>> , προσθέτοντάς της την θεωρία του Lazarus για το πώς ενεργοποιούνται αυτά τα συναισθήματα. Ουσιαστικά υπάρχει ένα σχήμα το οποίο μπορεί να εξηγήσει αυτή την θεωρία: Γεγονος(Χωρισμός από μια σχέση) · Γνωστική Αξιολόγηση(π.χ. Φταίει το άλλο άτομο γιατί με αδίκησε και με μείωνε κλπ.) · Πυρηνικό

Θέμα(π.χ. Η Αδικία) · Ενεργοποίηση του συναισθήματος(Π.Χ. Θυμός) · Συστατικά · (Αύξηση παλμών, Γρήγορη-κοφτή ανάσα, ανάληψη δράσης κλπ).

- 2) **Κατασκευαστική Προσέγγιση:** Σε αντίθεση με την πρώτη προσέγγιση η κατασκευαστική δεν παίρνει σαν δεδομένο την ύπαρξη βασικών και προ υπάρχοντων συναισθημάτων. Το συναίσθημα αναδύεται από την δυναμική αλληλεπίδραση διαφόρων παραγόντων και αποτελεί μια κατασκευή. Θεωρείται ότι το συναίσθημα δημιουργείται χωρίς να υπάρχει μια συγκεκριμένη αξιολόγηση και το αντίστροφο, η αξιολόγηση δεν είναι απαραίτητο ότι θα δημιουργήσει συναισθηματική αντίδραση.

2 Affective Computing

2.1 Τι είναι το Affective Computing

Στο παρόν κεφάλαιο θα γίνει μια εμπεριστατωμένη ανάλυση το κλάδου του Affective Computing. Από την απαρχή της ύπαρξης υπολογιστικών συστημάτων οι άνθρωποι, μέσω αυτών, προσπάθησαν να απαντήσουν σε πολύπλοκα προβλήματα και να προσομοιάσουν την ανθρώπινη συμπεριφορά. Ένα από αυτά τα προβλήματα είναι κατά πόσο και σε ποιο βαθμό οι υπολογιστές μπορούν να μιμηθούν την ανθρώπινη συμπεριφορά και αν και σε ποιο βαθμό μπορούν να αναπτύξουν συναισθήματα. Αν και το τελευταίο κομμάτι υπάγεται, προς το παρόν στον τομέα της επιστημονικής φαντασίας η μελέτη των συναισθημάτων και της σύνδεσής τους με τους υπολογιστές αποτελεί έναν βασικό κλάδο της τεχνητής νοημοσύνης.

Ο Υπολογιστικός συναισθηματισμός(Affective Computing) είναι εκείνο το κομμάτι της μηχανικής μάθησης που προσπαθεί να κατανοήσει, επεξεργαστεί, αναγνωρίσει και να προσομοιώσει το ανθρώπινο συναίσθημα μέσω ηλεκτρονικών υπολογιστών. Ο κλάδος αυτός πρώτο θεμελιώθηκε από τον Prof. Picard το 1997[6]. Στόχος ήταν η δημιουργία μιας ανθρωποκεντρικής τεχνητής νοημοσύνης, που θα κατανοούσε σε μεγαλύτερο βάθος την ανθρώπινη φύση και θα συνέβαλε στην θετική στην επίλυση σημαντικών προβλημάτων. Μερικά παραδείγματα είναι τα εξής: **(α)** Ένα λογισμικό το οποίο θα μπορεί με μεγάλη ακρίβεια να αναγνωρίζει 'real-time' τα συναισθήματα ενός ασθενή και να χρησιμοποιείται ενισχυτικά στην δουλειά ενός ψυχολόγου, έτσι ώστε ο γιατρός να μπορεί να διαμορφώνει με μεγαλύτερη ακρίβεια ένα ποιο κατάλληλο θεραπευτικό πλάνο. **(β)** Ένας Αι εικονικός βοηθός ο οποίος μέσω των εκφράσεων του προσώπου, της θερμοκρασίας του σώματος, των παλμών της καρδιάς κλπ, θα μπορεί να προσφέρει στο άτομο κάτι που θα του αλλάξει την διάθεση.

Τα δύο βασικά σημεία στα οποία μπορούμε να χωρίσουμε τον τομέα του Affective Computing είναι:

- **Αναγνώριση Συναισθήματος(Emotion Recognition):** Ουσιαστικά πρόκειται για λογισμικά τα οποία μπορούν να αναγνωρίσουν συναισθήματα μέσω των εκφράσεων του προσώπου, της φωνής, του γραπτού λόγου κλπ.

Παραδείγματα τέτοια είναι, τα Συνελικτικά Νευρωνικά Δίκτυα τα οποία μπορούν με μεγάλη ακρίβεια να προβλέψουν συναισθήματα μέσω εικόνων.

- **Ανάλυση Στάσης(Sentiment Analysis):** Η Ανάλυση Στάσης επικεντρώνει την προσοχή της στην αναγνώριση συναισθημάτων μέσω κειμένου. Πλέον υπάρχει μια μεγάλη ποικιλία από Μοντέλα Γλωσσικής Επεξεργασίας(NLP), όπως το Chat Gpt, DeepSeek, Gemini κλπ τα οποία μπορούν να κάνουν ανάλυση συναισθημάτων από τις λέξεις που χρησιμοποιεί το άτομο και από την δομή και το νόημα των προτάσεων που γράφει. Είναι πλέον πολύ διαδεδομένο αυτά τα μοντέλα να παίρνουν τον ρόλο του 'ψυχολόγου', με πολλά άτομα να στρέφονται για ψυχοθεραπεία στα NLP!

2.2 Emotion Models For Computers

Οι υπολογιστές δεν είναι σαν τους ανθρώπους. Δεν μπορούν να νιώσουν χαρά, λύπη, στεναχώρια, θυμό κλπ. Είναι μηχανές οι οποίες λαμβάνουν δεδομένα και αποκωδικοποιούν στην γλώσσα που καταλαβαίνουν, '0-1'. Οπότε να 'διδάξουμε' μια μηχανή να νιώθει είναι, προς το παρόν απίθανο. Αυτό που μπορούμε να κάνουμε είναι να της εισάγουμε μεγάλο αριθμό δεδομένων, με την μεγαλύτερη δυνατή καθαρότητα στην επεξεργασία του, ώστε το νευρωνικό δίκτυο να αρχίσει να μαθαίνει μέσω της διαδικασίας της εκπαίδευσης. Πέρα όμως από το πρακτικό κομμάτι που είναι η εκπαίδευση καθ' αυτή των μοντέλων με δεδομένα, υπάρχουν δύο κύριες φιλοσοφικές κατευθύνσεις που μετουσιώνουν την ποσότητα των δεδομένων σε ποιότητα για το υπολογιστή:

- **Διακριτά/Κατηγορικά Μοντέλα:** Κυρίως αυτό που χρησιμοποιείται και αυτό που χρησιμοποιώ στη δική μου εργασία. Τα δεδομένα ουσιαστικά, κατηγοριοποιούνται με διάφορες 'ταμπέλες'(Labels). Αυτή η φιλοσοφική προσέγγιση βασίζεται κατά κύριο λόγο στην θεωρία του Ekman για τα 6 βασικά συναισθήματα. Όπως προαναφέρθηκε ο Ekman χωρίζει τα συναισθήματα σε 6 βασικά(χαρά, λύπη, θυμός, αηδία, έκπληξη, φόβος), τα οποία χωρίζονται έτσι με το ότι 1) Προέρχονται από το ανθρώπινο ένστικτο. 2) Οι συνθήκες δεν παίζουν τόσο ρόλο στην εκδήλωσή τους, υπάρχει μια καθολικότητα ως προς την έκφραση των συγκεκριμένων συναισθημάτων. 3) Όταν βιώνουμε ένα συναίσθημα, ο τρόπος που βιώνεται δεν αλλάζει. Για παράδειγμα όταν νιώθουμε χαρά η καρδιά μας χτυπάει πιο γρήγορα, χαμογελάμε κλπ. 4) Τα συγκεκριμένα συναισθήματα εμφανίζουν τα ίδια χαρακτηριστικά

ανεξαρτήτως κουλτούρας, φυλής, φύλου, ηλικίας κλπ[6]. Πάνω στον Ekman πάτησε ο Plutchik για να εισάγει το δικό του μοντέλο, το 'τροχό συναισθημάτων', που περιλαμβάνει 8 βασικά συναισθήματα(χαρά, εμπιστοσύνη, φόβος, έκπληξη, λύπη, προσδοκία, θυμός, αηδία). Εστιάζει στο πώς αυτά συνδέονται μεταξύ τους .

- **Διαστασιακά/Συνεχή Μοντέλα:** Επειδή τα μοντέλα που έχουν τα διακριτά χαρακτηριστικά αναλύουν την πραγματικότητα με ιδιαίτερα μονοδιάστατο τρόπο, για να ξεπεραστεί αυτή η αδυναμία των συγκεκριμένων θεωριών αναπτύχθηκε η φιλοσοφία ότι τα συναισθήματα υπάρχουν και αλληλοσυνδέονται σε έναν πολυδιάστατο χώρο. Ποιο συγκεκριμένα, αναπτύχθηκε το μοντέλο PAD(Pleasure-Arousal-Dominance). α) Pleasure: Ο βαθμός ευχαρίστησης που νιώθουμε σε διάφορες καταστάσεις από την απλή χαρά μέχρι την έκσταση. β) Arousal: Ελέγχει το κατά πόσο είναι έτοιμο το άτομο σε ψυχολογικό επίπεδο, το level ετοιμότητάς του. γ) Dominance: Ο βαθμός ελέγχου των συναισθημάτων και η επιρροή που ασκεί το άτομο μέσω αυτών. Επειδή τα περισσότερα συναισθήματα υπάγονται στις κατηγορίες Pleasure και Arousal, αναπτύχθηκε το μοντέλο Valence-Arousal, στο οποίο το Valence μπαίνει στον άξονα των Χ και το Arousal στον άξονα των Υ. Συνδυάζοντας και τις δύο αυτές κατηγορίες το μοντέλο αυτό μπορεί να αποτυπώσει με ακρίβεια και ενδιάμεσες καταστάσεις. Για παράδειγμα, ζήλεια = αρνητικό Valance + υψηλό Arousal.

2.3 Affective Computing and Sentiment Analysis

Στο προηγούμενο κεφάλαιο αναφέρθηκε η θεωρητική βάση πάνω στην οποία τα υπολογιστικά μοντέλα αναγνωρίζουν και επεξεργάζονται τα συναισθήματα.

Τώρα θα εξεταστεί το πώς πρακτικά οι υπολογιστές επεξεργάζονται τα δεδομένα αυτά μετατρέποντάς τα σε συναισθήματα.

Οι δύο βασικοί πυλώνες του Affective Computing είναι το Emotion Recognition και το Sentiment Analysis. Το πρώτο εξετάστηκε στο προηγούμενο κεφάλαιο και αποτελεί την αναγνώριση των συναισθημάτων όπως είναι η χαρά, η λύπη, ο θυμός κλπ. Επειδή όμως η συγκεκριμένη προσέγγιση εμπεριέχει αρκετές προκλήσεις(Υποκειμενικότητα, Πολιτισμικές διαφορές κλπ.), έχει γίνει η σύνδεση μεταξύ Affective Computing και Sentiment Analysis.

Η Sentiment Analysis(Ανίχνευση Πολικότητας) εισάγει ένα πλαίσιο μέσα στο οποίο η αναγνώριση συναισθημάτων δεν γίνεται μέσα από το πρίσμα μεμονωμένων συναισθημάτων αλλά μέσω 'πόλων' πχ θετικό-αρνητικό. Το θέμα εδώ είναι από ένα συνδυασμό ερεθισμάτων να εξαχθεί ένα

συνολικότερο συμπέρασμα. Για παράδειγμα μοντέλα όπως το 'Hour Glass of Emotions' δείχνουν ότι η πολικότητα μπορεί να προκύψει μέσα από συνδυασμό συναισθημάτων, πχ ένα κείμενο με έντονη χαρά και αισιοδοξία είναι σαφώς 'θετικό'[7]. Αντίστοιχα όπως εξάγουμε συμπεράσματα για την πολικότητα σ' ένα κείμενο από τις λέξεις που χρησιμοποιούνται, τις φράσεις και τον γενικότερο τόνο έτσι και σε επίπεδο οπτικό χρειάζονται διάφορα ερεθίσματα για να αναλυθεί σωστά ένα συναίσθημα πχ έκφραση προσώπου, γλώσσα-στάση του σώματος κλπ.[7].

Σε πρακτικό επίπεδο υπάρχουν 3 προσεγγίσεις στο πως να εκπαιδεύσουμε ένα υπολογιστικό σύστημα να εντοπίζει την συναισθηματική κατάσταση ενός κειμένου πχ[7]

(α) Τεχνικές βασισμένες στην Γνώση(Knowledge-Based): Αυτή η προσέγγιση χρησιμοποιεί 'λεξικά γνώσης' τα οποία ανάλογα με την συχνότητα συγκεκριμένων λέξεων σ' ένα κείμενο μπορούν να εξάγουν συμπέρασμα για την πολικότητα του κειμένου. Για παράδειγμα αν η λέξη 'χαρά' εμφανίζεται 50 φορές και η λέξη 'αισιοδοξία' 20 φορές τότε το μοντέλο βγάζει το συμπέρασμα ότι το κείμενο έχει θετική πολικότητα. Το μοντέλο αυτό είναι αποτελεσματικό γιατί δεν χρειάζεται πολλά δεδομένα για να εξάγει συμπεράσματα και η χρήση του είναι σχετικά απλή. Παρόλα αυτά μειονεκτεί στην κατηγοριοποίηση συνθέτων φράσεων και στο ότι έχει σχετικά περιορισμένο λεξιλόγιο.

(β) Στατιστικές Μέθοδοι και Μηχανική Μάθηση: Εδώ η προσέγγιση που ακολουθείται βασίζεται σε κλασικά μοντέλα μηχανικής μάθησης ή νευρωνικά δίκτυα. Τα μοντέλα αυτά δέχονται πάρα πολλά δεδομένα, τα επεξεργάζονται και εξάγουν συμπεράσματα μέσω αυτών. Επειδή δέχονται πολλά δεδομένα και τα επεξεργάζονται, αναγνωρίζοντας πολύπλοκες συσχετίσεις, τα μοντέλα αυτά αποτελούν πανίσχυρα εργαλεία στην διαδικασία της ανάλυσης πολικότητας. Ταυτόχρονα με το θετική πλευρά της μεγάλης ποσότητας δεδομένων, υπάρχει και η αρνητική, τ' ότι το μοντέλο μπορεί να υπερεκπαιδευτεί πάνω σ' αυτά και να μην αναγνωρίζει νέα μοτίβα.

(γ) Συνδυασμός των δύο Μοντέλων: Με σκοπό να ενισχύσουμε τα πλεονεκτήματα και των δύο και να εξαλείψουμε τα μειονεκτήματά τους χρησιμοποιούμε τον συνδυασμό και των δύο. Για παράδειγμα θα μπορούσαμε να χρησιμοποιήσουμε ένα 'λεξικό γνώσης' και τις λέξεις που και τις λέξεις που βρίσκουμε να της τροφοδοτούμε σ' ένα νευρωνικό δίκτυο. Έτσι πετυχαίνουμε και την σημασιολογική σαφήνεια του πρώτου μοντέλου και την στατιστική ανάλυση του δεύτερου.

2.4 Sentiment Analysis and Neural Networks

Πέρα από την ανάλυση των συναισθημάτων όπως αναφέρθηκε εκτενέστατα στο προηγούμενο κεφάλαιο με Knowledge Based, Στατιστικές και Υβριδικές τεχνικές υπάρχει και η ανάλυση των συναισθημάτων μέσω εικόνων η οποία χρησιμοποιείται ως βοηθητικό εργαλείο σε στην ψυχολογία/ψυχιατρική, σε νευροαναπτυξιακές διαταραχές κλπ.

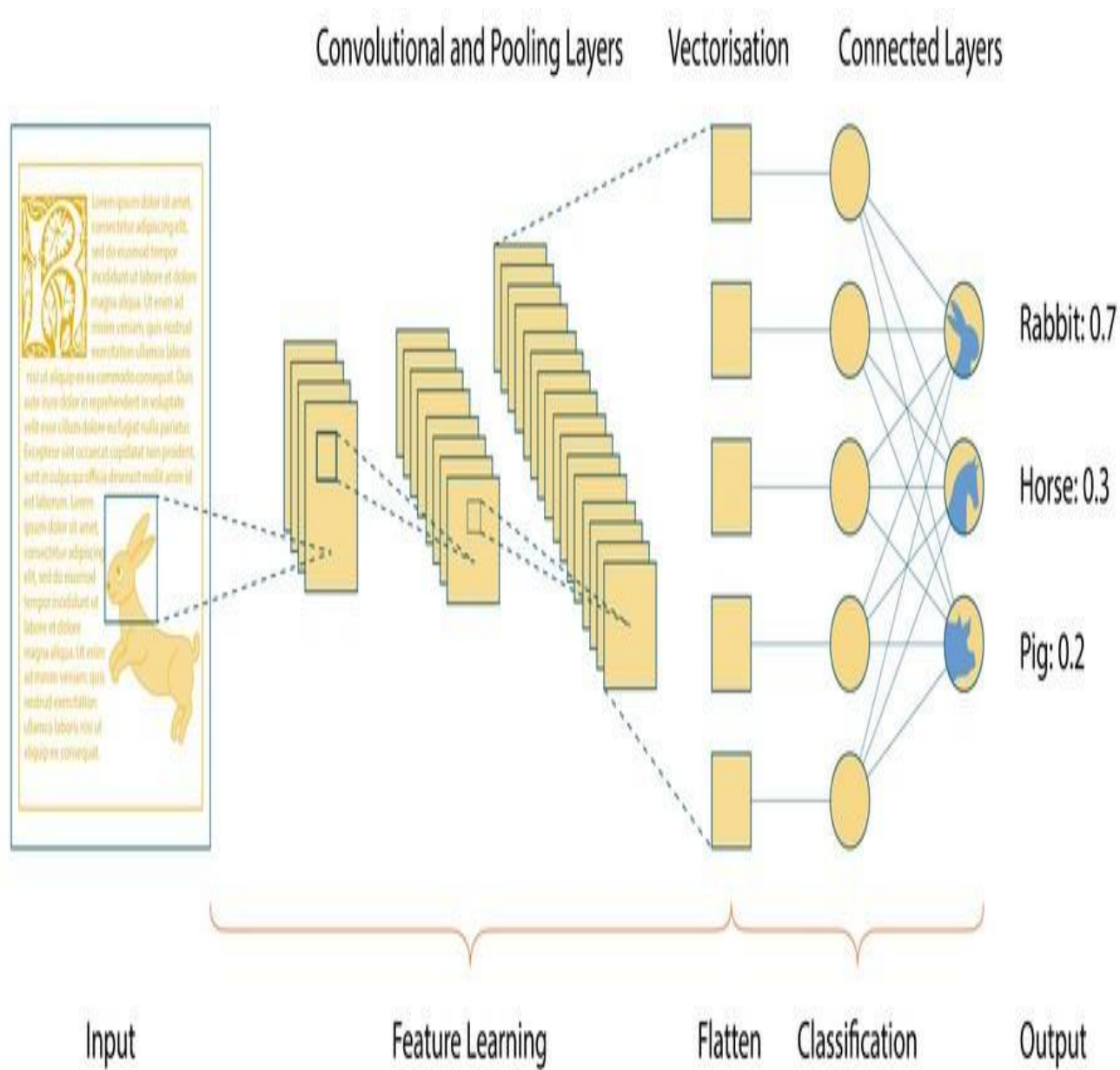
Η ανάλυση των εικόνων αυτών πριν την εισαγωγή στον τομέα αυτό του Deep Learning γινόταν 'χειροκίνητα' από τον άνθρωπο. Η διαδικασία ήταν η εξής[8]:

(α) Preprocessing: Προετοιμασία της εικόνας μέσω περικοπής περιπτών στοιχείων, αλλαγή στην φωτεινότητα, αλλαγές στις γωνίες κλπ.

(β) Feature Extraction: Η διαδικασία της εξαγωγής χαρακτηριστικών και της ανάλυσης τους γινόταν με προκαθορισμένους αλγορίθμους (Discrete Wavelet Transform και Linear Discriminant Analysis).

(γ) Classification: Η ταξινόμηση γινόταν με ταξινομητές όπως ο SVM και τα Random Forests.

Η διαδικασία αυτή παρόλο που ήταν πρωτοπόρα για την εποχή της βασιζόταν σε μεγάλο βαθμό στον ανθρώπινο παράγοντα για την εξαγωγή συμπερασμάτων και η μη αυτοματοποίηση πολλών διαδικασιών δημιουργούσε θέματα στον χρόνο εξαγωγής των αποτελεσμάτων. Η επανάσταση σ' αυτό τον τομέα ήρθε με την εισαγωγή των νευρωνικών δικτύων και ποιο συγκεκριμένα των Συνελικτικών Νευρωνικών δικτύων. Με αυτή την αλλαγή το Feature Extraction πλέον δεν χρειαζόταν αλγορίθμους που χρησιμοποιούνταν από τον άνθρωπο αλλά όλη η διαδικασία γινόταν αυτόματα από το νευρωνικό δίκτυο η εικόνες μελετώνται σε μεγαλύτερο βάθος και λεπτομέρεια και τα δεδομένα για εκπαίδευση μπορούν να είναι πάρα πολλά ενισχύοντας την διαδικασία της ταξινόμησης.



Πηγή:

https://commons.wikimedia.org/wiki/File:Convolutional_Neural_Network.png

3 Convolutional Neural Networks

3.1 Τι είναι τα Συνελικτικά Νευρωνικά Δίκτυα

Τα συνελικτικά νευρωνικά δίκτυα είναι αυτά τα οποία ουσιαστικά ήρθαν και έλυσαν το ζήτημα της εξαγωγής χαρακτηριστικών (Feature Extraction) από εικόνες και το Classification τους. Μια εικόνα η οποία είναι $256 \times 256 \times 3$, δηλαδή 256 Pixels και έγχρωμη(3), αποτελείται από 256 επί 256 επί 3 στοιχεία, δηλαδή 196.608 στοιχεία αν την κάνουμε flattened σε ένα διάνυσμα. Αυτός ο αριθμός δεδομένων είναι πάρα πολύ μεγάλος για να μπει σαν Input σ' ένα απλό νευρωνικό δίκτυο, ειδικά από την στιγμή που θέλουμε να φτιάξουμε ένα αναλυτικό νευρωνικό δίκτυο με πολλά Hidden Layers κλπ. Σε αυτή την περίπτωση τα στοιχεία που έχει να χειριστεί το δίκτυο μπορείς να γίνουν εκατομμύρια καθιστώντας την εκπαίδευσή του σχεδόν ανέφικτη.

Το παραπάνω πρόβλημα ήρθε να λύσει το Συνελικτικό Νευρωνικό Δίκτυο. Η ανακάλυψη τους οφείλεται στο πείραμα των Hubel & Wiesel το 1959. Αυτό το πείραμα το οποίο πραγματοποιήθηκε σε ζώα εξέταζε το πώς ενεργοποιούνται διαφορετικοί νευρώνες στον οπτικό φλοιό αναλόγως το οπτικό ερέθισμα που δέχεται το μάτι και ότι υπάρχουν 'ιεραρχικά επίπεδα' στο τι αντιλαμβάνονται αυτοί οι νευρώνες[9]. Αυτό το πείραμα έθεσε τις βάσεις και έδωσε το θεωρητικό πλαίσιο πάνω στο οποίο ο LeCun το 1990 ανέπτυξε το μοντέλο LeNet-5 το οποίο αποτελούταν από πάρα πολλά layers και δούλευε με Backpropagation Algorithm όπως και τα τυπικά νευρωνικά δίκτυα.

Τα Convolutional Neural Networks αποτελούνται από:

A) Convolutional Layers: Ουσιαστικά είναι η ραχοκοκαλιά των συνελικτικών νευρωνικών δικτύων. Χρησιμοποιούν φίλτρα ('kernels') για να μειώσουν τις διαστάσεις της εικόνας και να εξάγουν διαφορετικά συμπεράσματα.

B) Pooling Layers: Σε αντίθεση με τα Convolutional Layers που εξάγουν και συμπεράσματα τα Pooling Layers μειώνουν μόνο τον αριθμό των pixels μιας εικόνας.

Γ) Activation Functions: Όπως και σ' ένα απλό Fully Connected Neural Network έτσι και στα Convolutional Neural Networks χρησιμοποιείται η Relu function στο input layer και στα Hidden layers και η SoftMax function στο τελικό, output Layer.

Δ) Batch Normalization: Η διαδικασία αυτή ουσιαστικά 'κανονικοποιεί' τις ενεργοποιήσεις κάθε στρώματος με μέσο όρο 0 και διασπορά 1.

E) Callbacks: Όπως το Early Stopping και το ReduceLROnPlateau. Αυτά ουσιαστικά ελέγχουν και μπορεί να

σταματήσουν την διαδικασία αν το δίκτυο κάνει overfitting στα δεδομένα εκπαίδευσης.

Z) Fine-Tuning: Το μοντέλο επαναεκπαιδεύεται με τα δεδομένα που δέχεται να είναι αυτά από την πρώτη φάση εκπαίδευσης, συνήθως σε περισσότερες Epoch και με μικρότερο Learning Rate.

3.2 Διαφορές CNN'S με FCNN'S

Παρόλο που πολλά πράγματα στην αρχιτεκτονική τους είναι κοινά τα CNN'S έχουν καίριες διαφορές με τα FCNN'S τόσο στην δομή τους όσο και στο γιατί τα CNN'S είναι καταλληλότερα στο να μπορούν να αναλύσουν εικόνες. Οι κυριότερες απ' αυτές είναι:

A) Τα CNN's χρησιμοποιώντας τα kernels(πίνακες μικρών διαστάσεων 3×3 , 5×5 κλπ, τα οποία μετακινούνται κατά μήκος της κάθε εικόνας ανά βήμα που το ορίζουμε εμείς και 'σκανάρει' τα βασικά στοιχεία της εικόνας). Έτσι μια εικόνα η οποία στην αρχή αποτελείται από 256×256 pixels μετά το πρώτο convolutional layers μπορεί να μειωθεί στα 128×128 pixels κλπ. Αυτή η διαδικασία του Convolution σε συνδυασμό με το Max Pooling θα συνεχιστεί μέχρι να μείνει ένα τελικό διάνυσμα το οποίο ουσιαστικά θα 'ταϊστεί' σ' ένα Πλήρες συνδεδεμένο νευρωνικό δίκτυο.

B) Το CNN's εκμεταλλεύεται τις τοπικές γειτνιάσεις(Collaterations) και συσχετίσεις μεταξύ των pixels. Έχει διαπιστωθεί ότι τα γειτονικά pixels είναι πανομοιότυπα. Για παράδειγμα σε αυτήν την εικόνα:



Πηγή: <https://commons.wikimedia.org/w/index.php?search=cat&title=Special%3AMediaSearch&type=image>

Τα άσπρα pixels τα οποία συνθέτουν το χιόνι είναι το ένα δίπλα στο άλλο όπως γίνεται και στην πραγματική ζωή. Αντίστοιχα, όπως και τα pixels του τριχώματος της γάτας. Αυτό βοηθάει στο να αντιλαμβάνεται συσχετίσεις το δίκτυο μας μέσω των convolutional layers που δεν θα μπορούσε ένα FFN's να αντιληφθεί. Επίσης η αναγνώριση της γάτας πχ στην συγκεκριμένη περίπτωση δεν εξαρτάται από το που βρίσκεται στην φωτογραφία, καθώς η σύνθεση της φιγούρας 'γάτα' στην εικόνα έχει να κάνει με τις γειτνιάσεις των pixels. Αντίθετα σε πλήρως συνδεδεμένο νευρωνικό δίκτυο αν αλλάζαμε έστω και ελάχιστα την θέση της γάτας στην εικόνα το δίκτυο αυτό δεν έχει την ικανότητα να αντιληφθεί αυτή την αλλαγή οπότε θα κατέληγε σε εσφαλμένο συμπέρασμα.

3.3 Χαρακτηριστικά CNN's

Στην παρούσα ενότητα η οποία θα χωριστεί σε υπό ενότητες θα αναλυθούν τα βασικότερα χαρακτηριστικά που συνθέτουν ένα Συνελικτικό Νευρωνικό Δίκτυο. Όπως αναφέρθηκε και προηγουμένως πολλά στοιχεία είναι κοινά, καθώς όταν κάνουμε την διαδικασία μείωσης των διαστάσεων κάθε εικόνας με τα Convolutional Layers και το Max pooling, καταλήγουμε σ' ένα διάνυσμα τιμών το οποίο τροφοδοτείται στο σ' ένα Fnn's. Οπότε η ανάλυση μας θα επικεντρωθεί κυρίως στα στοιχεία εκείνα που τα ξεχωρίζουν από τα Πλήρως Συνδεδεμένα Νευρωνικά Δίκτυα. Τα στοιχεία αυτά είναι:

3.3.1 Convolutional Layers

Τα συνελικτικά στρώματα αποτελούν τον ακρογωνιαίο λίθο και το κύριο στοιχείο που διαφοροποιεί τα CNN's με τα FNN's. Ο τρόπος που λειτουργούν μοιάζει πολύ με την λειτουργία της όρασης στα έμβια όντα. Ουσιαστικά, με την χρήση φίλτρων(Kernels), τα οποία είναι πίνακες με 0 και 1 πολύ μικρών διαστάσεων πχ 3×3 , 5×5 κλπ σαρώνει ανά σειρά και ανάλογα με το βήμα που ορίζουμε εμείς(Stride = 1 σημαίνει ότι θα κινείται ανά ένα βήμα πάνω στην εικόνα). Κινούμενοι κατά μήκος της εικόνας υπολογίζουμε το Dot Product κάθε ξεχωριστού πίνακα ίδιων διαστάσεων, δημιουργώντας διαφορετικά Feature Maps τα οποία αποτελούνται από το άθροισμα των πινάκων που δημιουργήσαμε πριν συν το Bias. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να φτάσουμε σ' ένα Feature Map το οποίο να είναι ένα διάνυσμα τιμών, με το οποίο θα τροφοδοτήσουμε το FNN's μας. Αυτή η διαδικασία είναι αποτελεσματική επειδή τα convolutional layers εκμεταλλεύονται 3 χαρακτηριστικά:

A) Τοπικές Συσχετίσεις: Όπως αναφέρθηκε και προηγουμένως τα convolutional layers εκμεταλλεύονται τις συσχετίσεις μεταξύ των γειτονικών pixel. Αυτό σε επίπεδο πίνακα σημαίνει ότι τα μηδενικά θα είναι κοντά το ένα με το άλλο και αντίστοιχα οι άσσοι.

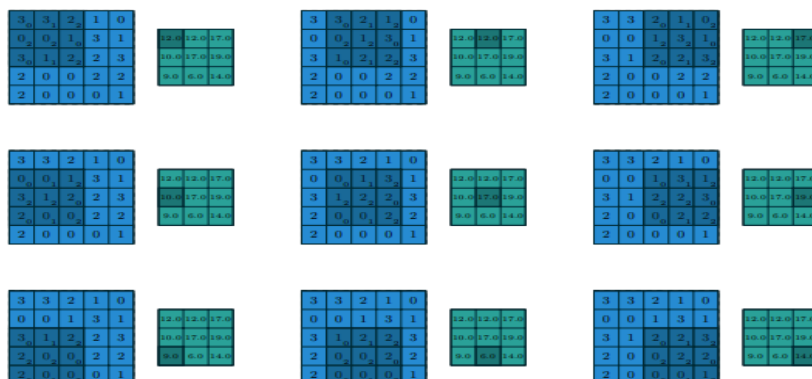
Β) Κοινή Χρήση Παραμέτρων: Χρησιμοποιεί τις ίδιες τιμές οι οποίες προκύπτουν από Back Propagation, τα ίδια βάρη και το ίδιο Bias. Έτσι μπορεί να αναγνωρίζει μοτίβα(άκρες, γωνιές κλπ) στην εικόνα ανεξάρτητα από την θέση τους.

Γ) Ιεραρχική αναπαράσταση: Δημιουργώντας πολλά convolutional layers τα οποία 'στοιβάζονται' το ένα πάνω στο άλλο δημιουργούμε διάφορα επίπεδα αναγνώρισης διαφορετικών στοιχείων. Για παράδειγμα, τα πρώτα στρώματα αναγνωρίζουν γωνιές, ακμές, χρώματα κλπ. Τα επόμενα αρχίζουν σιγά σιγά να 'βλέπουν' μοτίβα, έτσι αναγνωρίζουν σημεία του προσώπου κλπ. Τα τελευταία στρώματα είναι αυτά τα οποία φτάνουν σε βάθος την ανάλυση των εικόνων και μπορούν να αναγνωρίσουν πρόσωπα, εκφράσεις προσώπου αντικείμενα κλπ.

Για να δούμε ποιο αναλυτικά και στην πράξη πως δουλεύουν τα convolutional layers ας εξετάσουμε το πολύ γνωστό μοντέλο **LeNet-5[10]**:

- Το LeNet-5 δέχεται σαν Input εικόνες οι οποίες είναι 32*32 pixels.
- Το πρώτο layer αποτελείται από 6 feature maps 28*28 pixels και έχει 156 δεδομένα προς εκπαίδευση και 122.304 συνδέσεις.
- Το δεύτερο layer αποτελείται από 6 feature maps 14*14 pixels και έχει 12 δεδομένα προς εκπαίδευση και 5880 συνδέσεις.
- Το τρίτο layer αποτελείται από 16 feature maps και έχει 1516 δεδομένα προς εκπαίδευση και 156.000 συνδέσεις.
- Το τέταρτο layer αποτελείται από 6 feature maps 5*5 pixels και έχει 32 δεδομένα προς εκπαίδευση και 2000 συνδέσεις.
- Το Πέμπτο layer αποτελείται από 6 feature maps 1*1 pixels και έχει 48.120 δεδομένα προς εκπαίδευση.
- Το έκτο layer αποτελείται από 84 νευρώνες και έχει 10.164 δεδομένα στα οποία προσθέτονται τα βάρη, το bias και τα δεδομένα περνάνε από μια sigmoid function.

Τέλος για γίνει κατανοητή η διαδικασία που ακολουθούνε τα convolutional layers ακολουθεί μία εικόνα παρακάτω:



Πηγή: [11] A guide to convolution arithmetic for deep learning Vincent Dumoulin 1 F and Francesco Visin 2 F † FMILa, Université de Montréal †AIRLab, Politecnico di Milano

3.3.2 Max Pooling

Η διαδικασία του Max pooling αποτελεί μια θεμελιώδη λειτουργία των Συνελικτικών Νευρωνικών Δικτύων. Σε αντίθεση με τα convolutional layers, το max pooling επιτελεί μόνο την λειτουργία του down sampling στις εικόνες. Η δομή του είναι ίδια με τα με αυτή των convolutional layers. Χρησιμοποιεί έναν πίνακα μικρών διαστάσεων πχ 3*3 με stride 1 ή 2 κλπ, επιλέγει την μέγιστη τιμή στον πίνακα και την μεταβιβάζει στο feature map. Μερικά από τα πλεονεκτήματα της χρήσης του Max Pooling είναι:

- A) Μείωση Διαστάσεων:** Συμπληρωματικά με τα convolutional layers, το max pooling μειώνει δραματικά τις διαστάσεις των εικόνων, συμβάλλοντας στην καλύτερη διαχείριση των δεδομένων και στην αποφυγή του overfitting.
- B) Δεν επηρεάζεται από τις μετατοπίσεις:** Όπως και τα convolutional layers, έτσι και το max pooling δεν επηρεάζεται από την θέση του αντικείμενου προς εξέταση. Δηλαδή αν το αντικείμενο μετατοπιστεί αριστερά πχ πάλι θα αναγνωρισθεί.

Γ) Μεταφέρει τα πιο σημαντικά χαρακτηριστικά: Επιλέγοντας τον μεγαλύτερο αριθμό του πίνακα ουσιαστικά κληρονομεί στον feature map τα στοιχεία αυτά που έχουν την μεγαλύτερη βαρύτητα.

Για να γίνει κατανοητή η διαδικασία του Max Pooling παρακάτω δίνεται ένα παράδειγμα μ' έναν τυχαίο πίνακα:

❖ **Feature Map**

$$\begin{bmatrix} 1 & 6 & 8 & 9 \\ 5 & 3 & 9 & 1 \end{bmatrix}$$

[8,0,3,4]
[4,4,5,2]

Εφαρμογή max pooling 2*2 με stride=1:

- [[1,6],[5,3]] • max = 6
- [[6,8],[3,9]] • max = 9
- [[8,9],[9,1]] • max = 9
- [[5,3],[8,0]] • max = 8
- [[3,9],[0,3]] • max = 9
- [[9,1],[3,4]] • max = 9
- [[8,0],[4,4]] • max = 8
- [[0,3],[4,5]] • max = 5
- [[3,4],[5,2]] • max = 5

“Έξοδος

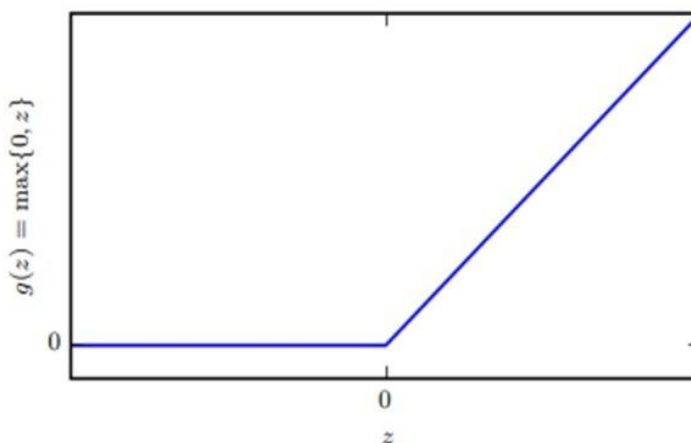
[[6,9,9]
[8,9,9]
[8,5,5]]

3.3.3 Activation Functions

Όπως και στα FNN's έτσι και στα Συνελικτικά Νευρωνικά Δίκτυα η χρήση των activation functions είναι κομβική για την σωστή λειτουργία του δικτύου. Η λειτουργία τους βασίζεται στο πως δουλεύουν οι νευρώνες στον ανθρώπινο εγκέφαλο. Όταν υπάρχει κάποιο οπτικό ερέθισμα για παράδειγμα πυροδοτούνται διάφοροι νευρώνες στον εγκέφαλο που φιλτράρουν την πληροφορία. Έρευνες έχουν δείξει ότι μόνο το 1 με 4 τα εκατό των νευρώνων μπορεί να είναι ταυτόχρονα ενεργοποιημένοι στον εγκέφαλο[12]. Αυτό το φαινόμενο ονομάζεται Αραιή Κωδικοποίηση. Με σκοπό να συμβαδίσει το μοντέλο μας μ' αυτή την αρχή τα δεδομένα φιλτράρονται με την ReLu(Rectified Linear Unit).

Η συνάρτηση ReLu, ορίζεται ως $f(x) = \max\{0, x\}$. Είναι η συνάρτηση που χρησιμοποιείται πιο συχνά στην διαδικασία του Forward Propagation[13]. Η λειτουργία της βασίζεται στον τρόπο που δουλεύει ο ανθρώπινος εγκέφαλος, πυροδοτεί μόνο τους νευρώνες για τους οποίους το $x > 0$ και για όσους $x < 0$ η τιμή της συνάρτησης γίνεται 0 απενεργοποιώντας ουσιαστικά τον νευρώνα.

Η γραφική της αναπαράσταση είναι η εξής:



Πηγή: [13] Goodfellow, I., Bengio, Y., & Courville, A. (2016). "Deep Learning." (Κεφάλαιο 6)-Σελ. 174-175

4. Dataset-Dataset Processing

4.1 Διαδεδομένα Datasets για Emotion Recognition

Όπως και στα FNN's έτσι και στα Συνελκτικά Νευρωνικά Δίκτυα το κυριότερο κομμάτι για την εξαγωγή σωστών συμπερασμάτων είναι το Dataset. Στην συγκεκριμένη περίπτωση μιλάμε για Dataset τα οποία αποτελούνται από εικόνες και χρησιμοποιούνται στην διαδικασία του Facial Recognition και Face Detection.

Οι εικόνες πρέπει να είναι ομοιόμορφες, να έχουν ίδιο μέγεθος σε pixel και ίδιο χρώμα(είτε γκρι, είτε έγχρωμο). Είναι απαραίτητο συνήθως να εφαρμοστεί είτε η μέθοδος του Oversampling ή του Undersampling, καθώς ο αριθμός των εικόνων σε κάθε φάκελο μπορεί να έχουν μεγάλη διαφορά. Σημαντικό επίσης είναι το Data Pre-Processing, ώστε να ενισχυθούν στοιχεία των εικόνων που θα μας βοηθήσουν να εξαγάγουμε τα συμπεράσματα που θέλουμε.

Μερικά από τα πιο διαδεδομένα dataset για Facial Recognition είναι:

- Από τα πιο γνωστά και τα μεγαλύτερα σε όγκο φωτογραφιών είναι τα Aff-Wild και Aff-Wild2. Και τα δύο αναπτύχθηκαν από τον D.Kollias. Το πρώτο εμπεριέχει 1.250.000 εικόνες που χωρίζονται σε 7 κατηγορίες με αριθμό pixels ανά εικόνα που κυμαίνεται κατά μέσο όρο 640*360 pixels. Το Aff-Wild2 αποτελείται από 2.800.000 φωτογραφίες οι οποίες χωρίζονται σε 7 κατηγορίες και ο μέσος όρος του μεγέθους τους είναι 1030*630[14].

- Ένα άλλο διαδεδομένο Dataset είναι το Labeled Faces in the Wild (LFW). Αποτελείται από 13.233 training data και 5749 test data. Έχει γίνει cropping στις φωτογραφίες καθώς το ζητούμενο εδώ είναι το face detection[14].
- Τέλος το FERET (Facial recognition technology) έχει 11.338 εικόνες από 1119 άτομα σε διάφορες 'πόζες'. Ιδανικό για classification, δεν έχει υποστεί pre-processing[14].

4.2 Επεξεργασία Δεδομένων

Τα δεδομένα αποτελούν τον ακρογωνιαίο λίθο για την σωστή εξαγωγή συμπερασμάτων. Τις περισσότερες φορές τα δεδομένα που έχουμε στην διάθεση μας δεν είναι κατευθείαν έτοιμα για εκπαίδευση. Μερικά από τα προβλήματα που μπορούμε να εντοπίσουμε στα datasets μας είναι τα εξής:

A) Λίγα Δεδομένα: Σε περίπτωση που τα δεδομένα προς εκπαίδευση είναι λίγα θα πρέπει να εφαρμόσουμε την μέθοδο του Oversampling για να αυξηθούν τα δεδομένα. Το νευρωνικό μας δίκτυο δεν θα εκπαιδευτεί σωστά με αποτέλεσμα να μην αναγνωρίζει εικόνες οι οποίες ειδικά διαφοροποιούνται από το training set μας.

B) Κακής Ποιότητας Δεδομένα: Δεδομένα τα οποία δεν περιέχουν τις σωστές πληροφορίες, πχ θολές εικόνες κλπ θα δημιουργήσουν πρόβλημα στην εκπαίδευση του δικτύου οδηγώντας σ' εσφαλμένα συμπεράσματα.

Γ) Πολλά Δεδομένα: Αν τα δεδομένα μας είναι πάρα πολλά το μοντέλο κινδυνεύει να υπερπροσαρμοστεί στα training data (overfitting) και να μην αναγνωρίζει άλλα δεδομένα.

Δ) Ελλείπουσες Τιμές: Σε κάθε dataset που έχουμε στην διάθεσή μας είναι σημαντικό να δούμε αν λείπουν τιμές σε σχέση με αυτές που αναγράφονται.

Τα δεδομένα έρχονται σε μορφή εικόνας, audio και βίντεο. Η μηχανή δεν μπορεί να αντιληφθεί την συγκεκριμένη γλώσσα. Οπότε θα πρέπει να τα μετατρέψουμε πρώτα σε αριθμητικά δεδομένα, ώστε ο υπολογιστής να μπορέσει να τα μεταφράσει στην γλώσσα του, σε 0 και 1 δηλαδή. Αυτή η διαδικασία ονομάζεται Normalization και οι πιο συνηθισμένες τεχνικές είναι[15]:

A) Min-Max Feature Scaling: Χρησιμοποιείται με σκοπό να φέρει όλες τις τιμές στο εύρος [0, 1][15].

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

B) Standard Score: Κανονικοποιεί τις αποκλίσεις[15].

$$\frac{x - \mu}{\sigma}$$

Γ) Studentized Residual: Κανονικοποιεί τα Residuals στην Regression Analysis[15].

$$\frac{\hat{\varepsilon}_I}{\hat{\sigma}_I} = \frac{x - \hat{\mu}_I}{\hat{\sigma}_I}$$

4.3 Data Augmentation

Μια καίρια διαδικασία στην προ-επεξεργασία των εικόνων είναι αυτή του Data Augmentation. Ουσιαστικά αυτό που κάνουμε είναι να μετασχηματίζουμε την εικόνα αλλάζοντας κάποια στοιχεία της (Φωτεινότητα, γωνίες, Σκιάσεις κλπ), μη αλλοιώνοντας παράλληλα την πληροφορία από την οποία θέλουμε να εξαγάγουμε συμπεράσματα. Επίσης η χρήση του βοηθάει στην αποφυγή του Overfitting και βελτιώνει το Accuracy[16].

Μερικές τεχνικές που χρησιμοποιούνται για την επεξεργασία των εικόνων είναι[17]:

- A) Flipping:** Συνήθως χρησιμοποιείται στον οριζόντιο άξονα και όχι τόσο στον κατακόρυφο. Βοηθάει το δίκτυο να εξαγει συμπεράσματα για τις εικόνες από διάφορες γωνίες.
- B) Color Space:** Ουσιαστικά πρόκειται για αλλαγές στο RGB, τα κανάλια που ορίζουν τις αποχρώσεις των εικόνων.
- C) Cropping:** Συμβάλει στην μείωση των διαστάσεων των εικόνων και στην αφαίρεση ενδεχόμενου 'θορύβου'.
- D) Translation:** Μετατοπίζει την εικόνα πάνω, κάτω, δεξιά και αριστερά, γεμίζοντας τον υπόλοιπο χώρο με 0.

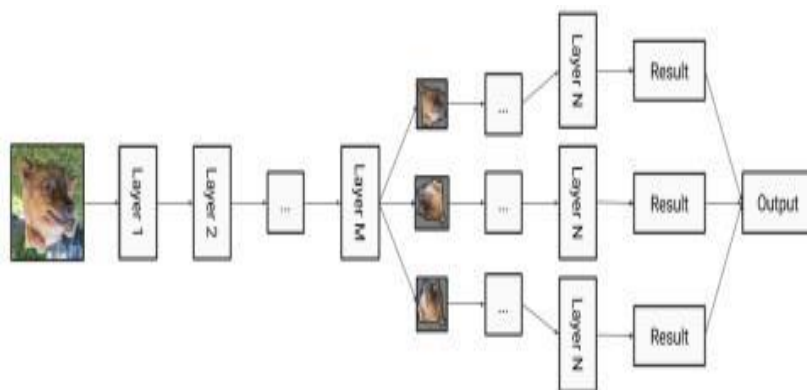


Fig. 1. Augmentation inside the network.

Πηγή: [16]

4.4 Class Imbalance

Στις περισσότερες περιπτώσεις τα δεδομένα που έχουμε στην διάθεσή μας θα είναι ανισομερώς κατανομημένα στις διάφορες κλάσεις που έχουμε. Αυτό αποτελεί καίριο πρόβλημα για την εξαγωγή σωστών συμπερασμάτων καθώς λόγω της μεγάλης ανισορροπίας το μοντέλο μας θα μεροληπτεί ως προς τα labels που έχουν περισσότερα δεδομένα. Για να λυθεί αυτό το ζήτημα μπορούμε να εφαρμόσουμε τρεις μεθόδους κατά την διάρκεια του του Data pre-processing.

- 1) **Oversampling:** Η τεχνική αυτή 'στοχεύει' τις κλάσεις με τα λιγότερα δεδομένα. Ουσιαστικά σκοπός της είναι να προσθέσει είδη υπάρχοντα δεδομένα, τυχαία από αυτά που βρίσκονται είδη στην κλάση ή μέσω της διαδικασίας του Data Augmentation παραλλάζοντας τα είδη υπάρχοντα δεδομένα να προσθέσει 'νέα'.
- 2) **Undersampling:** Η μέθοδος αυτή σε αντίθεση με το Oversampling στοχεύει τις κλάσεις με τα περισσότερα δεδομένα. Με σκοπό να αποφύγουμε την υπερπροσαρμογή του μοντέλου μας σε συγκεκριμένες κλάσεις, το οποίο θα οδηγήσει συνολικά το μοντέλο σε overfitting, αφαιρούμε τυχαία δεδομένα από εκείνες τις κλάσεις που υπερέχουν.
- 3) **Dynamic Sampling:** Χρησιμοποιήθηκε από τον Rouyanfar et al [18], αποτελεί συνδυασμός του oversampling, για τις κλάσεις που εμπεριέχουν πολλά δεδομένα και του undersampling για τις κλάσεις που στερούνται δεδομένα[18].

5. Experimental Design-Experimental Results

5.1 Dataset

5.1.1 Dataset Πριν τη Επεξεργασία

Το Dataset το οποίο χρησιμοποιήθηκε για την συγκεκριμένη εργασία είναι το Real-world Affective Faces Database (RAF-DB)[19]. Η συγκεκριμένη έκδοση του Dataset εμπεριέχει 11.285 εικόνες για training και 2906 εικόνες για validation. Οι εικόνες είναι της μορφής 100*100*3, δηλαδή 100*100 pixels και έγχρωμες(3). Αρχικά το dataset αποτελούνταν από 7 διαφορετικές κλάσεις: Happy, Sad, Disgust, Surprise, Neutral, Angry, Fear. Οι φάκελοι Angry, Disgust και Fear ενοποιήθηκαν σ' έναν φάκελο με το όνομα Intense Negative Emotions. Παρακάτω αναλύεται ο κώδικας που χρησιμοποιήθηκε για την ανάλυση του dataset πριν την επεξεργασία:

```
A) Έλεγχος Κατανομής των κλάσεων: print("2. ΕΛΕΓΧΟΣ ΚΑΤΑΝΟΜΗΣ
ΚΛΑΣΕΩΝ")

print("=" * 60)

class_counts = {}
total_images = 0

for emotion_id, emotion_name in emotion_map.items():
    emotion_path = os.path.join(train_path, emotion_name)
    image_files = [f for f in os.listdir(emotion_path)
                    if f.lower().endswith(('.png', '.jpg',
                    '.jpeg', '.bmp'))]
    class_counts[emotion_name] = len(image_files)
    total_images += len(image_files)

print(f"Σύνολο εικόνων στο training set: {total_images}")
print("\nΚατανομή εικόνων ανά κλάση:")
for emotion, count in class_counts.items():
    percentage = (count / total_images) * 100
    print(f"    - {emotion:12}: {count:>5} εικόνες
    ({percentage:5.1f}%)")
```

Με τον παραπάνω κώδικα ουσιαστικά βρίσκουμε τις κλάσεις του dataset, τον αριθμό των δεδομένων ανά κλάση και το ποσοστό των δεδομένων ανά κλάση. Τα αποτελέσματα ήταν τα εξής:

Κατανομή εικόνων ανά κλάση:
 - Happiness : 4772 εικόνες (42.3%)

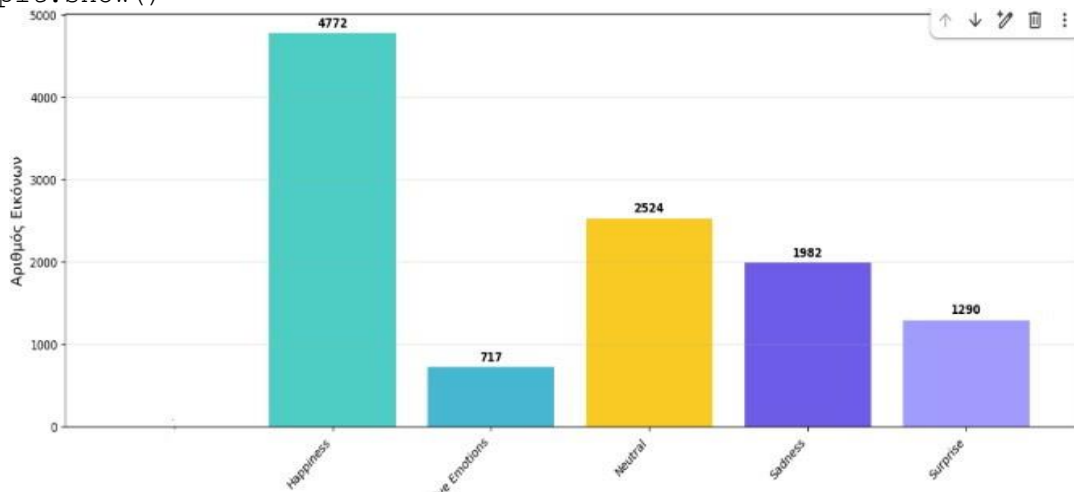
- Intense Negative Emotions: 717 εικόνες (6.4%)
- Neutral : 2524 εικόνες (22.4%)
- Sadness : 1982 εικόνες (17.6%)
- Surprise : 1290 εικόνες (11.4%)

Β) Οπτικοποίηση των αποτελεσμάτων:

```
lt.figure(figsize=(14, 7))
bars = plt.bar(range(len(class_counts)),
list(class_counts.values()),
color=['#ff6b6b', '#4ecdc4', '#45b7d1',
'#f9ca24', '#6c5ce7', '#a29bfe', '#fd79a8'])
plt.title('ΚΑΤΑΝΟΜΗ ΕΙΚΟΝΩΝ ΑΝΑ ΣΥΝΑΙΣΘΗΜΑ', fontsize=16,
fontweight='bold', pad=20)
plt.xlabel('Συναίσθημα', fontsize=12)
plt.ylabel('Αριθμός Εικόνων', fontsize=12)
plt.xticks(range(len(class_counts)),
list(class_counts.keys()), rotation=45, ha='right')

for i, bar in enumerate(bars):
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2., height + 50,
f'{int(height)}', ha='center', va='bottom',
fontweight='bold')

plt.grid(axis='y', alpha=0.3)
plt.tight_layout()
plt.show()
```



Γ) Έλεγχος Διαστάσεων και Εύρος Κλάσεων

```
shapes = set()
```



```

pixel_ranges = []

for emotion_id, emotion_name in emotion_map.items():
    emotion_path = os.path.join(train_path, emotion_name)
    image_files = [f for f in os.listdir(emotion_path)
                    if f.lower().endswith(('.png', '.jpg',
                                           '.jpeg', '.bmp'))]

    if image_files:

        for img_file in image_files[:3]:
            img_path = os.path.join(emotion_path, img_file)
            try:
                img = Image.open(img_path)
                img_array = np.array(img)
                shapes.add(img_array.shape)
                pixel_ranges.append((img_array.min(),
                                    img_array.max()))
            except:
                continue

print(f"Διαστάσεις εικόνων που βρέθηκαν: {shapes}")
print(f"Εύρος pixel values: {set(pixel_ranges)}")

```

Διαστάσεις εικόνων που βρέθηκαν: {(100, 100, 3)}

Εύρος pixel values: {(np.uint8(22), np.uint8(243)), (np.uint8(0), np.uint8(255)), (np.uint8(0), np.uint8(229)), (np.uint8(0), np.uint8(174)), (np.uint8(10), np.uint8(255)), (np.uint8(0), np.uint8(253)), (np.uint8(0), np.uint8(204)), (np.uint8(9), np.uint8(255))}

Δ) Οπτικοποίηση Δειγμάτων από κάθε κλάση

```

print("=" * 60)

plt.figure(figsize=(16, 10))
for i, (emotion_id, emotion_name) in
    enumerate(emotion_map.items()):
        emotion_path = os.path.join(train_path, emotion_name)
        image_files = [f for f in os.listdir(emotion_path)
                        if f.lower().endswith(('.png', '.jpg',
                                               '.jpeg', '.bmp'))]

        if image_files:

```

```

# Επιλογή πρώτης εικόνας
img_path = os.path.join(emotion_path, image_files[0])
img = Image.open(img_path)
img_array = np.array(img)

plt.subplot(2, 4, i + 1)
plt.imshow(img_array, cmap='gray')
plt.title(f'{emotion_name}\n{img_array.shape}',
fontweight='bold')
plt.axis('off')

plt.suptitle('ΔΕΙΓΜΑΤΑ ΕΙΚΟΝΩΝ ΑΠΟ ΚΑΘΕ ΣΥΝΑΙΣΘΗΜΑ',
fontsize=16, fontweight='bold')
plt.tight_layout()
plt.show()

```

Ε) Έλεγχος Corrupt Εικόνων

```

corrupt_count = 0
total_checked = 0

for emotion_id, emotion_name in emotion_map.items():
    emotion_path = os.path.join(train_path, emotion_name)
    image_files = [f for f in os.listdir(emotion_path)
                    if f.lower().endswith(('.png', '.jpg',
                    '.jpeg', '.bmp'))]

    for img_file in image_files[:10]:
        total_checked += 1
        img_path = os.path.join(emotion_path, img_file)

        try:
            img = Image.open(img_path)
            img.verify()
        except Exception as e:
            print(f"⚠ Corrupt εικόνα: {img_path}")
            corrupt_count += 1

print(f"Ελέχθηκαν {total_checked} εικόνες")
print(f"Βρέθηκαν {corrupt_count} corrupt εικόνες")

```

Δεν βρέθηκαν Corrupt Εικόνες.

5.1.2 Επεξεργασία Dataset

Επειδή το dataset μας έχει έντονο θέμα ανισορροπίας όσο αναφορά την αριθμητική κατανομή των δεδομένων, η κλάση Happiness έχει 4772 εικόνες, ενώ η κλάση Intense Negative Emotions έχει 717 εικόνες. Για να αντιμετωπιστεί αυτή η ανισορροπία εφαρμόζεται η μέθοδος του Oversampling. Παρακάτω παρουσιάζεται ο κώδικας και η ανάλυσή του:

A) Δημιουργία της συνάρτησης def create_precise_balanced_dataset

```
def create_precise_balanced_dataset(original_dir, new_dir,
target_sizes):

    if os.path.exists(new_dir):
        shutil.rmtree(new_dir)

    os.makedirs(new_dir, exist_ok=True)

    print("=== ΑΚΡΙΒΕΙΣ ΣΤΟΧΟΙ ===")
    for class_name, target_size in target_sizes.items():
        original_class_dir = os.path.join(original_dir,
class_name)
        original_size = len([f for f in
os.listdir(original_class_dir)
if f.endswith(('.jpg', '.png',
'.jpeg'))])
        print(f"{class_name}: {original_size} -> {target_size}
εικόνες")

    for class_name, target_size in target_sizes.items():
        print(f"\n--- ΕΠΕΞΕΡΓΑΣΙΑ: {class_name} ---")

        original_class_dir = os.path.join(original_dir,
class_name)
        new_class_dir = os.path.join(new_dir, class_name)
```

```

os.makedirs(new_class_dir, exist_ok=True)

        image_files      =      [f      for      f      in
os.listdir(original_class_dir)
        if f.endswith(('.jpg', '.png', '.jpeg'))]
current_size = len(image_files)

        print(f"Υπάρχουσες:      {current_size},      Στόχος:
{target_size}")

    if current_size >= target_size:

        for img_file in image_files[:target_size]:
            src = os.path.join(original_class_dir,
img_file)

            dst = os.path.join(new_class_dir, img_file)
            shutil.copy2(src, dst)
            print(f"✔ Αντιγράφηκαν {target_size} εικόνες")
        else:

            precise_oversampling(original_class_dir,
new_class_dir, current_size, target_size)

```

Δημιουργεί τους νέους φακέλους στους οποίους θα εφαρμοστεί το Oversampling

B) Δημιουργία της συνάρτησης def precise_oversampling

```

def precise_oversampling(original_dir, new_dir, current_size,
target_size):

    datagen = ImageDataGenerator (
        rotation_range=25,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True,
        brightness_range=[0.8, 1.2],
        fill_mode='nearest'
    )

```

```

image_files = [f for f in os.listdir(original_dir)
                if f.endswith(('.jpg', '.png', '.jpeg'))]

for img_file in image_files:
    src_path = os.path.join(original_dir, img_file)
    dst_path = os.path.join(new_dir, img_file)
    shutil.copy2(src_path, dst_path)

print(f"Αντιγράφηκαν {len(image_files)} αρχικές εικόνες")

needed = target_size - current_size

if needed <= 0:
    return

print(f"Δημιουργία {needed} νέων εικόνες...")

created_count = 0
image_index = 0

while created_count < needed:

    img_file = image_files[image_index % len(image_files)]
    image_index += 1

    try:

        img_path = os.path.join(original_dir, img_file)
        img = load_img(img_path)
        img_array = img_to_array(img)
        img_array = np.expand_dims(img_array, axis=0)

        base_name = os.path.splitext(img_file)[0]
        save_prefix =

        f'{base_name}_aug_{created_count:04d}'

        aug_iter = datagen.flow(
            img_array,

```

```

        batch_size=1,
        save_to_dir=new_dir,
        save_prefix=save_prefix,
        save_format='jpg'
    )

    next(aug_iter)
    created_count += 1

    if created_count % 100 == 0:
        print(f"      Δημιουργήθηκαν
{created_count}/{needed} εικόνες")

    except Exception as e:
        print(f"Σφάλμα στην εικόνα {img_file}: {e}")
        continue

    print(f"✔ Δημιουργήθηκαν {created_count} νέες εικόνες")

```

Εδώ ορίζεται ο στόχος των εικόνων που θέλουμε να φτάσουμε στον κάθε φάκελο. Εφαρμόζεται συνδυαστικά το oversampling με εικόνες που υπάρχουν είδη στις κλάσεις και εικόνες των κλάσεων που υπόκεινται σε επεξεργασία μέσω Data Augmentation.

Τέλος τα αποτελέσματα του τελικού dataset είναι τα εξής:

```

target_sizes = {
    'Happiness': 4772,          # Παραμένει ως έχει
    'Neutral': 2524,           # Παραμένει ως έχει
    'Sadness': 1982,           # Παραμένει ως έχει
    'Surprise': 3800,          # Από 1290 σε 3800
    'Intense Negative Emotions': 3500 # Από 717 σε 3500
}

```

5.2 Σχεδιασμός Συνελικτικού Δικτύου

Παρακάτω παρουσιάζετε και αναλύεται ο κώδικας που αναπτύχθηκε για τον σχεδιασμό του Συνελικτικού μου Δικτύου:

```

A) import tensorflow as tf
from tensorflow.keras import layers, models, optimizers

```

```

from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.callbacks import EarlyStopping,
ReduceLROnPlateau, ModelCheckpoint
import numpy as np
import matplotlib.pyplot as plt
from sklearn.utils import class_weight
import os
from google.colab import drive
import math

# Mount Google Drive
drive.mount('/content/drive')

# Δημιουργία φακέλου για τα models
os.makedirs('/content/drive/MyDrive/emotion_models',
exist_ok=True)

TARGET_SIZE = (100, 100)
BATCH_SIZE = 48
EPOCHS = 100

print("Δημιουργία Data Generators.")
train_dir = '/content/sample_data/TrainRaf_Balanced_Correct1'
val_dir = '/content/sample_data/TestRaf'

```

Γίνεται import των απαραίτητων βιβλιοθηκών, ορίζεται το target size, το batch size και οι epochs του μοντέλου και δημιουργούνται τα data generators για το training και το validation.

B) #Data Augmentaion για TrainRaf

```

train_datagen =
tf.keras.preprocessing.image.ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.15,
    height_shift_range=0.1,
    shear_range = 0.1,
    horizontal_flip=True,
    zoom_range=0.15,
    brightness_range=[0.85, 1.15],
    fill_mode='nearest'
)

```

```

)

#Data Augmentation για TestRaf
val_datagen =
tf.keras.preprocessing.image.ImageDataGenerator(
    rescale=1./255
)

#Train Generator
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=TARGET_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    shuffle=True,
    color_mode='rgb'
)

#Validation Generator
val_generator = val_datagen.flow_from_directory(
    val_dir,
    target_size=TARGET_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    shuffle=False,
    color_mode='rgb'
)

print(f"Train classes: {train_generator.class_indices}")
print(f"Validation classes: {val_generator.class_indices}")
print(f"Train samples: {train_generator.samples}")
print(f"Validation samples: {val_generator.samples}")

#Class-Weights
print("Υπολογισμός Class-Weights")
train_labels_sparse = train_generator.classes

class_weights = class_weight.compute_class_weight(
    'balanced',
    classes=np.unique(train_labels_sparse),
    y=train_labels_sparse

```



```

)
class_weights_dict = dict(enumerate(class_weights))
print(f"Class weights: {class_weights_dict}")

# Υπολογισμός steps per epoch για σωστό decay_steps
steps_per_epoch = math.ceil(train_generator.samples /
BATCH_SIZE)
print(f"Steps per epoch: {steps_per_epoch}")

```

Εδώ ορίζεται το data augmentation για το training και το validation dataset, ορίζουμε τα στοιχεία στα dictionary για το training και validation. Τέλος χρησιμοποιείται η μέθοδος του class-weight για την αποτελεσματικότερη αντιμετώπιση του class imbalance και υπολογίζονται τα steps per epoch για να έχουμε το ιδανικό decay_steps.

```

Γ) #Model Sequential
print("Δημιουργία Μοντέλου.....")

#Input Phase
model = tf.keras.Sequential()
model.add(tf.keras.Input(shape=(100, 100, 3)))

#First Conv2D and Max Pooling
model.add(tf.keras.layers.Conv2D(32, (3, 3),
activation='relu', padding='same'))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Conv2D(32, (3, 3),
activation='relu', padding='same'))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.MaxPooling2D((2, 2)))
model.add(tf.keras.layers.Dropout(0.25))

#Second Conv2D and Max Pooling
model.add(tf.keras.layers.Conv2D(64, (3, 3),
activation='relu', padding='same'))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Conv2D(64, (3, 3),
activation='relu', padding='same'))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.MaxPooling2D((2, 2)))
model.add(tf.keras.layers.Dropout(0.3))

```

```

#Third Conv2D and Max Pooling
model.add(tf.keras.layers.Conv2D(128, (3, 3),
activation='relu', padding='same'))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Conv2D(128, (3, 3),
activation='relu', padding='same'))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.MaxPooling2D((2, 2)))
model.add(tf.keras.layers.Dropout(0.3))

#Fourth Conv2D and Max Pooling
model.add(tf.keras.layers.Conv2D(256, (3, 3),
activation='relu', padding='same'))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.MaxPooling2D((2, 2)))
model.add(tf.keras.layers.Dropout(0.4))

#Classifier
model.add(tf.keras.layers.GlobalAveragePooling2D())
model.add(tf.keras.layers.Dense(512, activation='relu'))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Dropout(0.5))
model.add(tf.keras.layers.Dense(256, activation='relu'))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Dropout(0.4))
model.add(tf.keras.layers.Dense(5, activation='softmax'))

```

Δημιουργούμε το συνελκτικό δίκτυο το οποίο αποτελείται από input phase, 4 blocks με 2 convolutional layers το καθένα, 1 max pooling, batch normalization για το κάθε convolutional layer και ένα dropout layer. Το τελευταίο block είναι ο classifier αποτελείται από ένα dense με GlobalAveragepooling2D, 2 denses με activation function relu, 2 batch normalization layers, 2 dropout layers και το τελικό dense layer με activation function softmax.

```

Δ) # Learning Rate Schedule
initial_learning_rate = 0.001
total_steps_phase1 = steps_per_epoch * 30

print(f" Decay steps για CosineDecay: {total_steps_phase1}
(για 30 epochs)")

```

```

lr_schedule = tf.keras.optimizers.schedules.CosineDecay(
    initial_learning_rate=initial_learning_rate,
    decay_steps=total_steps_phase1,
    alpha=0.0001
)

print(f"✓ To LR θα πάει από {initial_learning_rate} σε 0.0001
σε 30 epochs")

#Model-Compile
model.compile(
    optimizer=optimizers.Adam(
        learning_rate=lr_schedule,
        beta_1=0.9,
        beta_2=0.999,
        epsilon=1e-07
    ),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

```

Ορίζουμε το αρχικό learning rate που θα χρησιμοποιηθεί και τα steps per epoch. Με το framework cosine decay δημιουργούμε το dictionary το οποίο θα ελέγχει τον ρυθμό με τον οποίο μειώνεται το learning rate. Τέλος γίνεται compile το μοντέλο με το Adam optimizer.

E) #Callbacks

```

early_stopping = EarlyStopping(
    monitor='val_accuracy',
    patience=25,
    restore_best_weights=True,
    mode='max',
    verbose=1
)

# ΒΕΛΤΙΩΜΕΝΟ ReduceLRonPlateau
reduce_lr = ReduceLRonPlateau(
    monitor='val_accuracy',
    factor=0.5,           # Μείωση 50%
    patience=8,           # 8 epochs χωρίς βελτίωση
    min_lr=0.000001,      # Ελάχιστο LR

```

```

        mode='max',
        verbose=1,
        cooldown=3,                # 3 epochs χωρίς αλλαγές μετά από
        μείωση
        min_delta=0.001            # Ελάχιστη βελτίωση για να θεωρηθεί
        βελτίωση
    )

    # ΒΕΛΤΙΩΜΕΝΑ CALLBACKS ΜΕ ΑΠΟΘΗΚΕΥΣΗ ΣΤΟ DRIVE
    checkpoint = ModelCheckpoint(
        '/content/drive/MyDrive/emotion_models/best_model.h5',
        monitor='val_accuracy',
        save_best_only=True,
        mode='max',
        verbose=1
    )

    # Backup κάθε 5 epochs
    backup_checkpoint = ModelCheckpoint(

        '/content/drive/MyDrive/emotion_models/backup_epoch_{epoch:0
        2d}.h5',
        save_freq=5*len(train_generator), # Κάθε 5 epochs
        save_weights_only=False,
        verbose=0
    )

    # Emergency backup αν σταματήσει το training
    emergency_backup = tf.keras.callbacks.LambdaCallback(
        on_epoch_end=lambda epoch, logs:
        model.save(f'/content/drive/MyDrive/emotion_models/emergency
        _epoch_{epoch}.h5')
    )

    # Λίστα με όλα τα callbacks
    callbacks_list = [early_stopping, reduce_lr, checkpoint,
        backup_checkpoint, emergency_backup]

    #Model-Fit1
    print("=" * 60)
    print("Ξεκινάει η εκπαίδευση - Φάση 1 (30 epochs)")

```

```

print(f"📌 Αρχικό LR: {initial_learning_rate}")
print(f"📌 Θα μειωθεί σε 0.0001 σταδιακά")
print("=" * 60)

history_phase1 = model.fit(
    train_generator,
    epochs=30,
    validation_data=val_generator,
    class_weight=class_weights_dict,
    callbacks=callbacks_list,
    verbose=1
)

print("Η πρώτη φάση της εκπαίδευσης ολοκληρώθηκε")
model.summary()

```

Εδώ εφαρμόζονται τα callbacks, το early stopping που σταματάει το μοντέλο όταν Val accuracy δεν βελτιώνεται μέχρι ένα συγκεκριμένο αριθμός epochs, το ReduceLROnPlateau το οποίο ελέγχει και μειώνει το learning rate ανάλογα με τις ανάγκες του δικτύου και τα διάφορα checkpoints που αποθηκεύουν το την φάση της εκπαίδευσης με το μεγαλύτερο validation accuracy. Τέλος γίνεται fit το μοντέλο για πρώτη φορά με τα train και Val generator που έχουμε ορίσει, με 30 epoch, το class weight και τα callbacks που έχουν οριστεί.

Z) #Fine-Tuning

```

print("=" * 60)
print("Recompile με μικρότερο learning rate για fine-tuning")
print("📌 Νέο LR: 0.0001")
print("=" * 60)

model.compile(
    optimizer=optimizers.Adam(learning_rate=0.0001),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

# Επαναφορά του ReduceLROnPlateau για τη δεύτερη φάση
reduce_lr_phase2 = ReduceLROnPlateau(
    monitor='val_accuracy',
    factor=0.5,
    patience=10,

```

```

        min_lr=0.000001,
        mode='max',
        verbose=1,
        cooldown=5
    )

    # Νέα λίστα callbacks για τη δεύτερη φάση
    callbacks_phase2 = [early_stopping, reduce_lr_phase2,
                        checkpoint, backup_checkpoint, emergency_backup]

    #Model-Fit2
    print("Συνέχεια Εκπαίδευσης - Φάση 2 (Fine-tuning)")
    history_phase2 = model.fit(
        train_generator,
        epochs=EPOCHS,
        validation_data=val_generator,
        class_weight=class_weights_dict,
        callbacks=callbacks_phase2,
        verbose=1
    )

    #ΣΥΝΔΥΑΣΜΟΣ ΙΣΤΟΡΙΚΩΝ
    history = {
        'accuracy': history_phase1.history['accuracy'] +
        history_phase2.history['accuracy'],
        'val_accuracy': history_phase1.history['val_accuracy'] +
        history_phase2.history['val_accuracy'],
        'loss': history_phase1.history['loss'] +
        history_phase2.history['loss'],
        'val_loss': history_phase1.history['val_loss'] +
        history_phase2.history['val_loss']
    }

    #ΑΠΟΤΕΛΕΣΜΑΤΑ
    print("Αποτελέσματα εκπαίδευσης:")

```

Σ' αυτό το σημείο είναι το fine tuning του μοντέλου. Αλλάζοντας το ReduceLROnPlateau, καθώς το ξανακάνουμε fit με μικρότερο learning rate και αλλάζοντας τις epochs από 30 σε 100, ξαναεκπαιδεύουμε τ μοντέλο εκεί που τελείωσε στην πρώτη φάση.

5.3 Αποτελέσματα

Στην πρώτη φάση της εκπαίδευσης τρέχουμε το μοντέλο σε 30 epochs αρχικό learning rate 0,001 το οποίο μειώνεται με το decay steps και το ReduceLROnPlateau. Η εκπαίδευση της πρώτης φάσης ολοκληρώνεται στην 30^η epoch με τα αποτελέσματα τα 27^η epoch να είναι αυτά που αποθηκεύονται στο τέλος με τα Val_accuracy 0.78596, accuracy 0.8108, loss 0.5466, Val_loss 0.7352 και learning_rate 2.4569e-05.

Οι Παράμετροι που χρησιμοποιήθηκαν κατά την πρώτη φάση της εκπαίδευσης είναι:

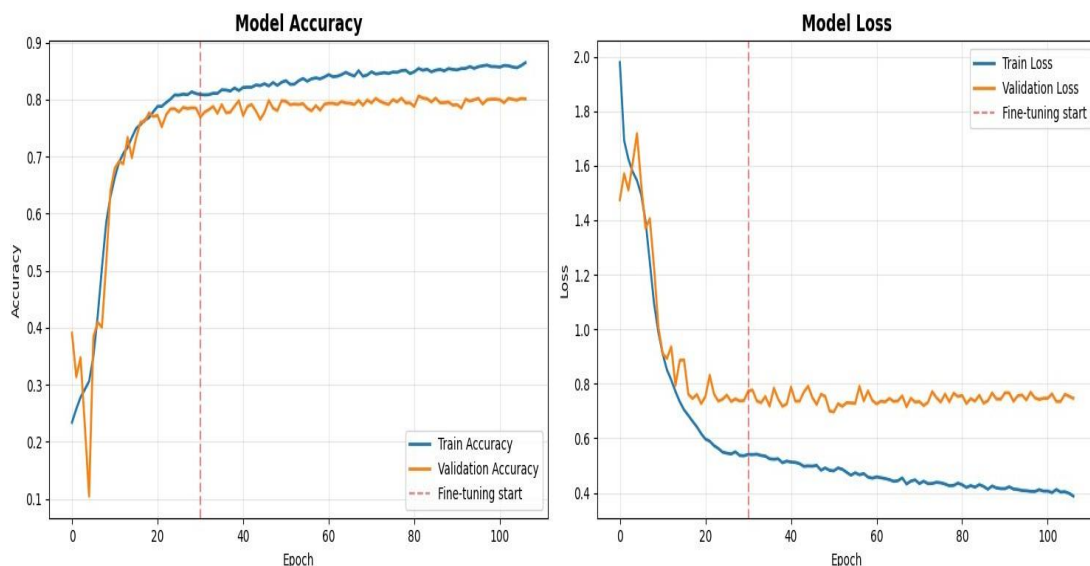
Total params: 2,550,896 (9.73 MB)

Trainable params: 849,317 (3.24 MB)

Non-trainable params: 2,944 (11.50 KB)

Optimizer params: 1,698,635 (6.48 MB)

Στην δεύτερη φάση γίνεται recompile του μοντέλου με learning rate 0,0001 το οποίο ελέγχεται από νέο ReduceLROnPlateau. Η εκπαίδευση αυτή την φορά τρέχει σε 100 epochs. Με το early stopping η εκπαίδευση σταματάει στην epoch 77, καθώς το validation accuracy δεν βελτιώνεται για 25 epoch από την epoch 52. Στην 52 αποθηκεύονται και τα τελικά αποτελέσματα του μοντέλου με Val_accuracy 0.8059, Train_accuracy 0.8645 και Test_loss 0.7275.



6. Συμπεράσματα και Μελλοντικές Προεκτάσεις

6.1 Συμπεράσματα

Με βάση τα αποτελέσματα το μοντέλο συνολικά έχει πετύχει τον σκοπό του με τελικό validation accuracy 0.8059. Αυτό πρακτικά σημαίνει ότι από της 100 εικόνες που θα του δώσουμε της 80 θα της εντάξει στην σωστή κλάση. Φαίνεται ότι μοντέλο έχει μάθει

σωστά τα διαφορετικά μοτίβα των εικόνων και δεν έχει υπερπροσαρμοστεί στα δεδομένα της εκπαίδευσης.

Στην πρώτη φάση της εκπαίδευσης το validation accuracy αυξάνεται συνεχώς και το loss πέφτει φτάνοντας στο μέγιστο στην 27^η epoch στο 0.78596. Ένα εξαιρετικό αποτέλεσμα για μόνο 27 epochs, δείχνοντας ότι ο συνδυασμός oversampling και class weight έχει δουλέψει πολύ καλά για την αντιμετώπιση του class imbalance. Επιπλέον έχει γίνει σωστό data augmentation και ο συνδυασμός του decay_steps με το ReduceLROnPlateau κρατώντας το learning rate σε ιδανικό επίπεδο ώστε να οδηγήσει σ' αυτό το αποτέλεσμα για το πρώτο training session.

Στην δεύτερη φάση της εκπαίδευσης το validation accuracy αυξάνεται πολύ λίγο, φτάνοντας την τιμή 0.8059. Στην συγκεκριμένη training session οι epochs που χρησιμοποιούνται είναι 100. Φαίνεται ότι το μοντέλο έχει πετύχει τόσο μεγάλο val_accuracy στην πρώτη φάση που στην 2^η μαθαίνει μόνο συγκεκριμένες λεπτομέρειες.

6.2 Μελλοντικές Προεκτάσεις

Με σκοπό να επιτευχθεί ακόμα πιο μεγάλη ακρίβεια και να μπορούν να χρησιμοποιηθούν και datasets με πολλά περισσότερα δεδομένα αυτές θα μπορούσαν να ήταν κάποιες μελλοντικές προεκτάσεις του παρόντος project:

A) Transformer Models

Ο τρόπος που δουλεύουν προσομοιάζει με τον τρόπο που δουλεύει το Nlp. Αυτό επιτρέπει στο δίκτυο να κάνει και συνδέσεις μεταξύ πχ των μερών του προσώπου σ' αντίθεση με τα convolutional layers. Τέλος μπορεί να χρησιμοποιηθεί για μεγάλο όγκο δεδομένων.

B) Deep Convolutional Neural Networks

Όπως για παράδειγμα το DenseNet, το οποίο η διαφορά του μ' ένα συμβατικό συνελικτικό νευρωνικό δίκτυο είναι ότι όλα τα layers συνδέονται μεταξύ τους, προσδίδοντας βάθος στο δίκτυο.

Γ) Transfer Learning

Η συγκεκριμένη μέθοδος μας δίνει την δυνατότητα να χρησιμοποιήσουμε dataset με πάρα πολλά δεδομένα. Συνδυάζοντας το νευρωνικό δίκτυο που έφτιαξα με ένα pre trained model πετυχαίνουμε καλύτερα αποτελέσματα.

6.3 Τελικές Παρατηρήσεις

Με βάση το dataset που είχα στην διάθεση μου και την υπολογιστική ισχύ του google collabatory, το τελικό project παρουσίασε ένα πολύ θετικό αποτέλεσμα χρησιμοποιώντας μια σχετικά απλή αρχιτεκτονική.

Η εκπαίδευση στην πρώτη της φάση 'έτρεξε' ιδανικά φτάνοντας σ' ένα πολύ μεγάλο validation accuracy μόλις στην 27^η epoch. Στην φάση του fine tuning φάνηκε ως ένα βαθμό η αδυναμία του dataset και των μεθόδων που χρησιμοποιήθηκαν, καθώς υπήρξε ελάχιστη αύξηση του val_accuracy.

Συνολικά αποτελεί μια σημαντική εισαγωγή στα συνελκτικά νευρωνικά δίκτυα τόσο σε θεωρητικό επίπεδο, καθώς παρουσιάστηκε και το θεωρητικό υπόβαθρο των συγκεκριμένων δικτύων, αλλά και σε πρακτικό επίπεδο, γιατί χρησιμοποιήθηκαν όλες οι βασικές και καίριες τεχνικές και το αποτέλεσμα ήταν αρκετά καλό.

6.4 Συγκεντρωτικός Πίνακας

Παρακάτω ακολουθούν οι πίνακες με τα τελικά αποτελέσματα:

Από την πρώτη φάση της Εκπαίδευσης

Layer(type)	Output Shape	Param #
conv2d_21 (Conv2D)	(None, 100, 100, 32)	896
batch_normalization_27 (BatchNormalization)	(None, 100, 100, 32)	128
conv2d_22 (Conv2D)	(None, 100, 100, 32)	9,248
batch_normalization_28 (BatchNormalization)	(None, 100, 100, 32)	128
max_pooling2d_12 (MaxPooling2D)	(None, 50, 50, 32)	0
dropout_18 (Dropout)	(None, 50, 50, 32)	0
conv2d_23 (Conv2D)	(None, 50, 50, 64)	18,496
batch_normalization_29 (BatchNormalization)	(None, 50, 50, 64)	256
conv2d_24 (Conv2D)	(None, 50, 50, 64)	36,928
batch_normalization_30 (BatchNormalization)	(None, 50, 50, 64)	256
max_pooling2d_13 (MaxPooling2D)	(None, 25, 25, 64)	0
dropout_19 (Dropout)	(None, 25, 25, 64)	0
conv2d_25 (Conv2D)	(None, 25, 25, 128)	73,856
batch_normalization_31 (BatchNormalization)	(None, 25, 25, 128)	512
conv2d_26 (Conv2D)	(None, 25, 25, 128)	147,584
batch_normalization_32 (BatchNormalization)	(None, 25, 25, 128)	512
max_pooling2d_14	(None, 12, 12, 128)	0

(MaxPooling2D)		
dropout_20 (Dropout)	(None, 12, 12, 128)	0
conv2d_27 (Conv2D)	(None, 12, 12, 256)	295,168
batch_normalization_33 (BatchNormalization)	(None, 12, 12, 256)	1,024
max_pooling2d_15 (MaxPooling2D)	(None, 6, 6, 256)	0
dropout_21 (Dropout)	(None, 6, 6, 256)	0
global_average_pooling2d_3 (GlobalAveragePooling2D)	(None, 256)	0
dense_9 (Dense)	(None, 512)	131,584
batch_normalization_34 (BatchNormalization)	(None, 512)	2,048
dropout_22 (Dropout)	(None, 512)	0
dense_10 (Dense)	(None, 256)	131,328
batch_normalization_35 (BatchNormalization)	(None, 256)	1,024
dropout_23 (Dropout)	(None, 256)	0
dense_11 (Dense)	(None, 5)	1,285

- **Total params:** 2,550,896 (9.73 MB)
- **Trainable params:** 849,317 (3.24 MB)
- **Non-trainable params:** 2,944 (11.50 KB)
- **Optimizer params:** 1,698,635 (6.48 MB)

Πίνακας με τα τελικά αποτελέσματα

7.

Test Accuracy	0.8059
Test Loss	0.7275
Καλύτερο val_accuracy	0.8059
Καλύτερο train_accuracy	0.8645
Σύνολο epochs	107
Τελικό test_accuracy	0.8059

Βιβλιογραφία

[1] Mordka, C. (2016). What are emotions? Structure and function of emotions. *Studia Humana*, 5(3), 29–44. <https://doi.org/10.1515/sh-2016-0013>

[2] Larue, O., West, R., Rosenbloom, P. S., Dancy, C. L., Samsonovich, A. V., Petters, D., & Juvina, I. (2018). Emotion in the common model of cognition.

Procedia Computer Science, 145, 740–746.
<https://doi.org/10.1016/j.procs.2018.11.061>

[3] Vaida, S., & Calbaza Ormenışan, M. (2013). From Plato to Ellis: A short investigation of the concept of emotions. *Procedia – Social and Behavioral Sciences*, 78, 571–575. <https://doi.org/10.1016/j.sbspro.2013.04.356>

[4] Ortony, A. (2022). Are all “basic emotions” emotions? A problem for the (basic) emotions construct. *Perspectives on Psychological Science*, 17(1), 216–232. <https://doi.org/10.1177/1745691620985415>

[5] Moors, A. (2014). Flavors of appraisal theories of emotion. *Emotion Review*, 6(4), 303–307. <https://doi.org/10.1177/1754073914534477>

[6] Wang, Y., Song, W., Tao, W., Liotta, A., Yang, D., Li, X., Gao, S., Sun, Y., Ge, W., Zhang, W., & Zhang, W. (2022). A systematic review on affective computing: Emotion models, databases, and recent advances. *Information Fusion*, 83–84, 19–52. <https://doi.org/10.1016/j.inffus.2022.03.009>

[7] Cambria, E. (2016). Affective computing and sentiment analysis. *IEEE Intelligent Systems*, 31(2), 102–107. <https://doi.org/10.1109/MIS.2016.31>

[8] Meena, G., Mohbey, K. K., & Kumar, S. (2023). Sentiment analysis on images using convolutional neural networks based Inception-V3 transfer learning approach. *International Journal of Information Management Data Insights*, 3(1), Article 100164. <https://doi.org/10.1016/j.jjime.2023.100164>

[9] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, 77, 354–377. <https://doi.org/10.1016/j.patcog.2017.10.034>

[10] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324. <https://doi.org/10.1109/5.726791>

[11] Dumoulin, V., & Visin, F. (2016). *A guide to convolution arithmetic for deep learning* [arXiv preprint]. arXiv:1603.07285. <https://arxiv.org/abs/1603.07285>

[12] Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the 14th International Conference on Artificial*

Intelligence and Statistics (pp. 315–323). JMLR Workshop and Conference Proceedings. <https://proceedings.mlr.press/v15/glorot11a.html>

[13] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning* (Κεφάλαιο 6, σελ. 174–175). MIT Press.

[14] List of datasets in computer vision and image processing. (2024). In *Wikipedia*.

https://en.wikipedia.org/wiki/List_of_datasets_in_computer_vision_and_image_processing#Facial_recognition

[15] Normalization (statistics). (2024). In *Wikipedia*.
[https://en.wikipedia.org/wiki/Normalization_\(statistics\)](https://en.wikipedia.org/wiki/Normalization_(statistics))

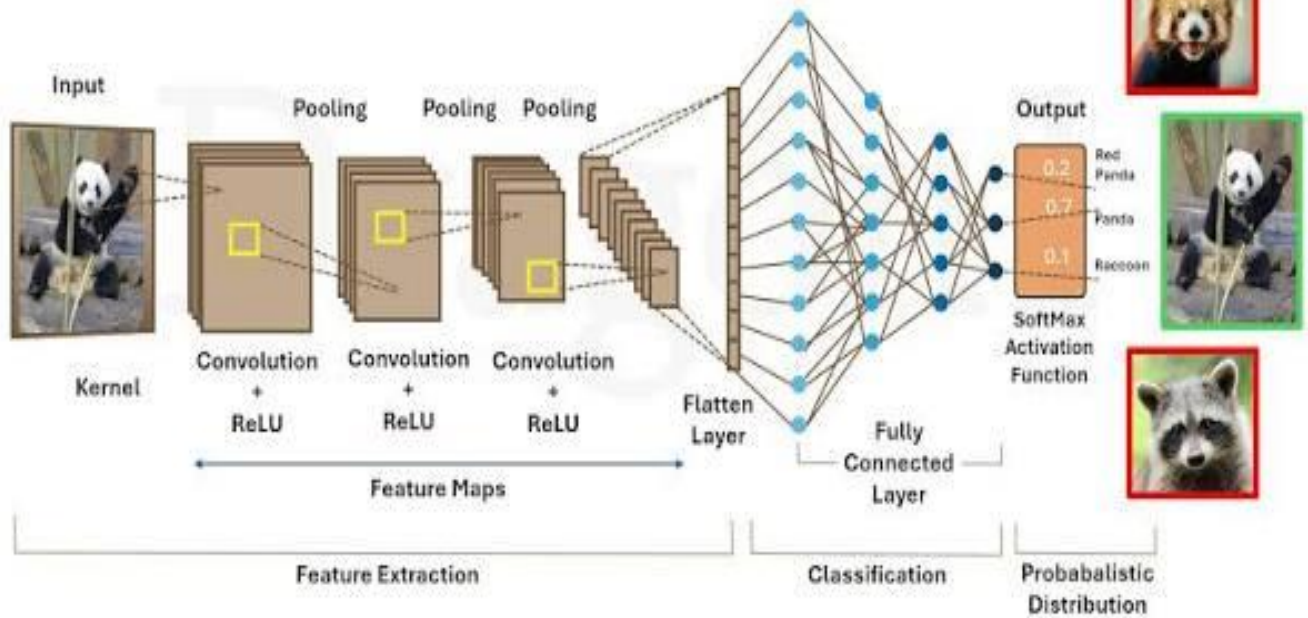
[16] Sypetkowski, M., Jasiulewicz, J., & Wojna, Z. (2020). *Augmentation inside the network* [arXiv preprint]. arXiv:2012.10769.
<https://arxiv.org/abs/2012.10769>

[17] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), Article 60.
<https://doi.org/10.1186/s40537-019-0197-0>

[18] Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1), Article 27.
<https://doi.org/10.1186/s40537-019-0192-5>

[19] Shuvoalok. (n.d.). *RAF-DB dataset* [Dataset]. Kaggle.
<https://www.kaggle.com/datasets/shuvoalok/raf-db-dataset>

Convolutional Neural Networks (AI Deep Learning)



© Copyright 2025, Dragon1, www.dragon1.com

Πηγή: <https://www.dragon1.com/terms/convolutional-neural-networks-definition>