Τμήμα Ηλεκτρολόγων Μηχανικών
& Μηχανικών Υπολογιστών
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΕΛΟΠΟΝΝΗΣΟΥ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Αυτοματοποιημένη Ανίχνευση και Ανάλυση Βίντεο DeepFake: Μια Προσέγγιση με Βαθιά μάθηση

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Ακριώτη Αναστασίας

Επιβλέπων: Ταμπακάς Βασίλειος, Καθηγητής
Συνεπιβλέπων: Πιντέλας Εμμανουήλ, Μεταδιδακτορικός Ερευνητής

Πάτρα 2025

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Πάτρα, 2025

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. ΒΑΣΙΛΕΙΟΣ ΤΑΜΠΑΚΑΣ
2. ΠΑΝΑΓΙΩΤΗΣ ΑΛΕΦΡΑΓΚΗΣ
3. ΣΩΤΗΡΙΟΣ ΧΡΙΣΤΟΔΟΥΛΟΥ

Υπεύθυνη Δήλωση Φοιτητή

Βεβαιώνω ότι είμαι συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τη συγκεκριμένη εργασία.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Πελοποννήσου δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω, πρωτίστως, τον επιβλέποντα καθηγητή μου, κ. Ταμπακά και τον συνεπιβλέποντα μεταδιδακτορικό ερευνητή κ. Πιντέλα, για τις πολύτιμες συμβουλές και την άμεση υποστήριξη τους κατά τη συγγραφή αυτής της εργασίας. Επίσης, την οικογένειά μου για τη στήριξή της όλα αυτά τα χρόνια, την εμπιστοσύνη και την ενθάρρυνσή της σε κάθε μου βήμα. Τέλος τους φίλους μου που μου στάθηκαν δίπλα μου σε όλη τη διάρκεια των φοιτητικών μου χρόνων.

Ακριώτη Αναστασία

Περίληψη

Το φαινόμενο των Deepfake βίντεο έχει εξελιχθεί σε μια σοβαρή και τεχνολογική απειλή, καθώς χρησιμοποιείται για τη διάδοση ψευδών ειδήσεων, την παραπληροφόρηση και τη χειραγώγηση της κοινής γνώμης. Οι προηγμένες τεχνικές σύνθεσης και επεξεργασίας πολυμεσικού υλικού καθιστούν την ανίχνευση τέτοιων παραποιημένων βίντεο ιδιαίτερα δύσκολη, γεγονός που δημιουργεί την ανάγκη για αξιόπιστες λύσεις. Στην παρούσα διπλωματική εργασία προτείνεται μια μέθοδος ανίχνευσης Deepfake περιεχομένου μέσω τεχνικών βαθιάς μάθησης, με στόχο τη βελτίωση της αποτελεσματικότητας στην ταξινόμηση εικόνων ως γνήσιες ή παραποιημένες. Για την υλοποίηση, χρησιμοποιείται το προεκπαιδευμένο συνελικτικό νευρωνικό δίκτυο EfficientNet, το οποίο εφαρμόζεται σε στατικές εικόνες που εξάγονται από βίντεο μέσω δειγματοληψίας. Η ανάλυση βασίζεται σε τρία διαφορετικά σύνολα δεδομένων, το καθένα από τα οποία διαθέτει ξεχωριστά χαρακτηριστικά, με σκοπό να διασφαλιστεί η γενικευσιμότητα του μοντέλου. Η ανάπτυξη του ταξινομητή πραγματοποιείται στη γλώσσα προγραμματισμού Python, αξιοποιώντας σύγχρονες βιβλιοθήκες βαθιάς μάθησης, ενώ η αξιολόγησή του βασίζεται σε μετρικές απόδοσης, όπως η ακρίβεια. Τα πειραματικά αποτελέσματα στην ανίχνευση Deepfake εικόνων αναδεικνύουν τη δυνατότητα χρήσης συνελικτικών νευρωνικών δικτύων ως ένα αποτελεσματικό εργαλείο για την αντιμετώπιση του προβλήματος.

Η συμβολή της παρούσας μελέτης έγκειται όχι μόνο στην τεχνική προσέγγιση της ανίχνευσης Deepfake περιεχομένου, αλλά και στην ανάδειξη της σοβαρότητας του προβλήματος, υπογραμμίζοντας την ανάγκη για εξελιγμένα μέσα προστασίας απέναντι στη διάδοση ψεύδους οπτικοακουστικού υλικού.

Λέξεις Κλειδιά

deepfake, convolutional neural network, deep learning, efficientNet

Abstract

Deepfake video has become a serious and technological threat, as it is used to spread fake news, misinformation, and manipulate public opinion. Advanced techniques for synthesizing and editing multimedia content make the detection of such manipulated videos particularly challenging, highlighting the need for reliable detection solutions. This thesis proposes a Deepfake detection method based on deep learning techniques, aiming to improve the accuracy of classifying images as real or fake. The implementation leverages the pre-trained convolutional neural network EfficientNet, which is applied to static frames extracted from video through sampling. The analysis is conducted on three distinct datasets, each with unique characteristics, to ensure the generalizability of the model. The classifier is developed using the Python programming language and modern deep learning libraries, its performance is evaluated using metrics such as accuracy. The experimental results demonstrate the potential of convolutional neural networks as an effective tool for addressing the deepfake detection problem.

The contribution of this study lies not only in its technical approach to detecting deepfake content but also in emphasizing the severity of the issue, highlighting the need for advanced mechanisms to protect against the spread of fake audiovisual material.

Keywords

deepfake, convolutional neural network, deep learning, efficientNet

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Σχημάτων

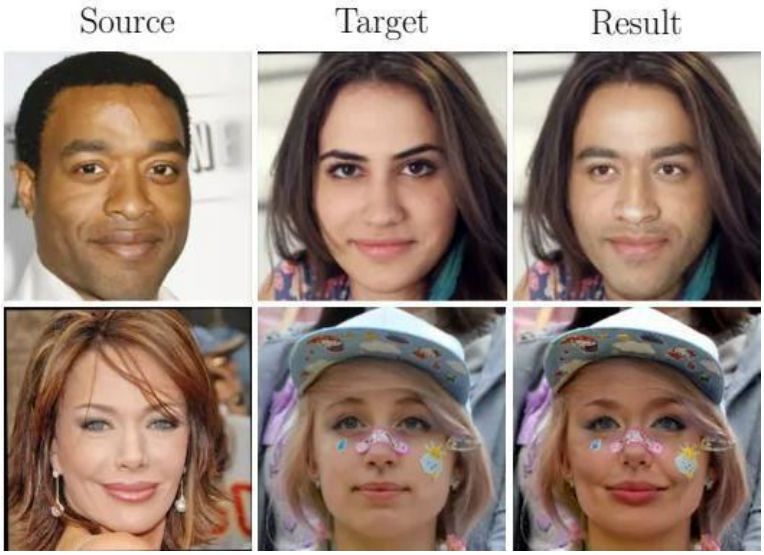
Κατάλογος Πινάκων

1. Εισαγωγή

1.1 Ορισμός και Προέλευση των Deepfakes

Τα τελευταία χρόνια οι ψευδείς ειδήσεις εξαπλώνονται ραγδαία μέσω των κοινωνικών δικτύων αποτελώντας σοβαρή απειλή για τον δημόσιο λόγο και την ανθρώπινη κοινωνία. Η άνοδος των λεγόμενων Deepfakes έχει επιδεινώσει την εξάπλωση ψευδών ειδήσεων, καθώς επιτρέπει τη δημιουργία πειστικού συνθετικού περιεχομένου βίντεο μέσω αλγορίθμων βαθιάς μάθησης. Τα deepfakes είναι προϊόν της τεχνητής νοημοσύνης που τροποποιεί εικόνες, βίντεο και ηχητικά μέσα, δημιουργώντας ψεύτικο περιεχόμενο. [1] Ο όρος DeepFake (DF) προέρχεται από το «Deep Learning (DL)» που σημαίνει βαθιά μάθηση και τη λέξη «fake» που σημαίνει ψεύτικο. Αυτός ο όρος πρωτοεμφανίστηκε στα τέλη του 2017, όταν ένας ανώνυμος χρήστης στην πλατφόρμα Reddit εφάρμοσε αυτές τις μεθόδους για να αντικαταστήσει το πρόσωπο ενός ατόμου σε βίντεο με ερωτικό περιεχόμενο, χρησιμοποιώντας το πρόσωπο ενός άλλου, με σκοπό τη δημιουργία ενός ψεύτικου βίντεο. [2]

Τα βίντεο deepfake διακρίνονται σε τρεις κατηγορίες. Η πρώτη τεχνική ονομάζεται «puppet-master», στην οποία η έκφραση του προσώπου και οι κινήσεις του κεφαλιού ενός ατόμου (master) χαρτογραφούνται σε ένα άλλο άτομο (puppet). Η δεύτερη κατηγορία χρησιμοποιεί την τεχνική «face-swap», η οποία περιλαμβάνει την αντικατάσταση του προσώπου ενός ατόμου με ένα άλλο, διατηρώντας όμως τις εκφράσεις και τις κινήσεις του προσώπου του αρχικού ατόμου. Η τρίτη κατηγορία βίντεο deepfake είναι το «lip sync». Δηλαδή, η δημιουργία ενός παραποιημένου βίντεο όπου τροποποιείται μόνο η κίνηση των χειλιών του ατόμου, έτσι ώστε να φαίνεται ότι λέει κάτι που δεν είπε στην πραγματικότητα. Αυτό επιτυγχάνεται με την ανακατασκευή των χειλιών του ατόμου, ώστε να συγχρονίζονται με τον ήχο ενός διαφορετικού διαλόγου ή ομιλίας. [3]



Εικόνα 1.1: Παράδειγμα της τεχνικής «face-swap».

Η δημιουργία ψεύτικου περιεχόμενου δεν είναι κάτι καινούριο, ωστόσο οι σύγχρονες εξελίξεις στην τεχνητή νοημοσύνη (Artificial Intelligence – AI) και τη βαθιά μάθηση έχουν αυξήσει την τάση των deepfakes. Το φαινόμενο του σχηματισμού πιστών παραποιημένων διαδικτυακών δημιουργιών αυξάνεται υψηλά στις πλατφόρμες κοινωνικών δικτύων με πολλές παραποιημένες εικόνες και πληθώρα βίντεο διασημοτήτων, πολιτικών, και διάσημων προσωπικοτήτων. Ο κίνδυνος και οι κοινωνικές επιπτώσεις είναι σημαντικές και πολύ επιζήμιες, ιδίως με την ελάχιστη τεχνική επάρκεια και τις συσκευές που απαιτούνται για την παραγωγή deepfakes. Τέτοιο περιεχόμενο μπορεί να δημιουργηθεί αβίαστα από οποιονδήποτε που έχει πρόσβαση στο διαδίκτυο. [1]
Τα deepfakes μπορούν πλέον να μιμούνται πειστικά τις εκφράσεις του προσώπου, τις κινήσεις των χειλιών, ακόμη και τις φωνές, καθιστώντας τα σχεδόν αδιάκριτα από τα πραγματικά βίντεο. Αυτές οι δυνατότητες κάνουν την εν λόγω τεχνολογία εξαιρετικά επικίνδυνη, δημιουργώντας μια αυξανόμενη ανάγκη για αποτελεσματικές τεχνικές ανίχνευσης για την αντιμετώπιση της κατάχρησής της. Η ανίχνευση deepfake είναι δύσκολη λόγω της πολυπλοκότητας των σύγχρονων τεχνικών σύνθεσης με γνώμονα την τεχνητή νοημοσύνη. Οι παραδοσιακές μέθοδοι ανίχνευσης, όπως αυτές που βασίζονται στην οπτική ποιότητα ή τις ασυνέπειες, καθίστανται λιγότερο αποτελεσματικές καθώς εξελίσσεται η deepfake τεχνολογία. [4]

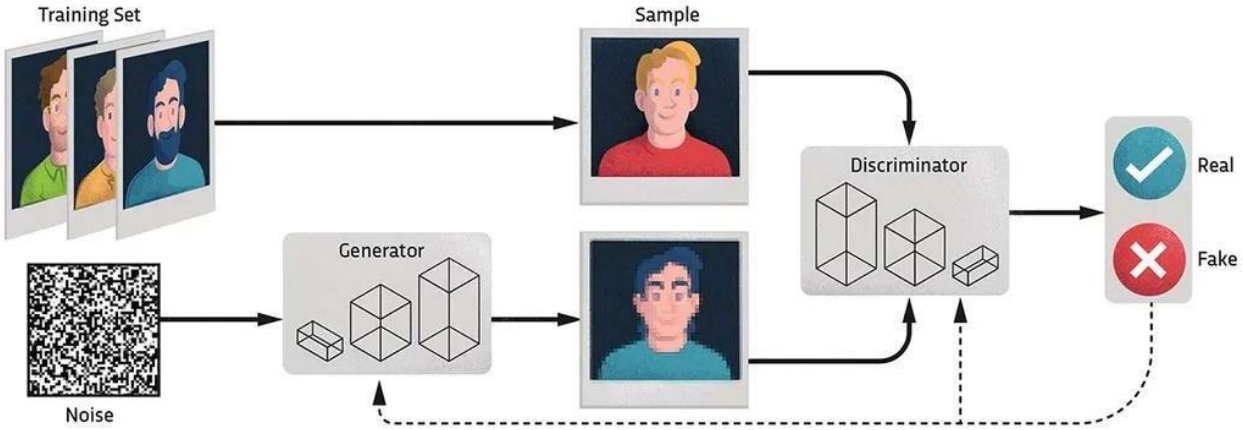
Η αύξηση της δημοτικότητας του βίντεο υπογραμμίζει την ανάγκη για εργαλεία που επιβεβαιώνουν την αυθεντικότητα του περιεχομένου των μέσων ενημέρωσης και των ειδήσεων, καθώς οι νέες τεχνολογίες επιτρέπουν την πειστική χειραγώγηση του βίντεο. Η διάδοση ψευδών πληροφοριών είναι εύκολη και θέτει σοβαρά ηθικά και νομικά ζητήματα, ωστόσο η ανάπτυξη αποτελεσματικών μεθόδων ανίχνευσης και η καταπολέμηση των deepfakes είναι πιο δύσκολη. Προκειμένου να επιτευχθεί αυτό, πρέπει να κατανοήσουμε πλήρως την τεχνολογία πίσω από αυτά, τους λόγους ύπαρξής τους και τις μεθόδους ανίχνευσής τους. [5] Η κατανόηση της τεχνολογίας deepfake, των επιπτώσεών της και των τρόπων αντιμετώπισής της είναι ζωτικής σημασίας για τη διατήρηση της ακεραιότητας της πληροφόρησης και της ασφάλειας στην ψηφιακή εποχή.

1.2 Η Τεχνολογία Πίσω από τα Deepfakes

Με την πρόοδο της τεχνητής νοημοσύνης και της υπολογιστικής όρασης, ο πιο αποτελεσματικός τρόπος για τη δημιουργία deepfakes βασίζεται σε τεχνικές μηχανικής μάθησης και συγκεκριμένα στα παραγωγικά αντιπαραθετικά δίκτυα (GANs), τα οποία με τη σειρά τους χρησιμοποιούν τη δομή των συνελικτικών νευρωνικών δικτύων (CNNs). Μια μεγάλη ποικιλία αλγορίθμων DeepFake που χρησιμοποιούν GANs έχουν προταθεί με σκοπό την αναπαραγωγή των εκφράσεων και των κινήσεων του προσώπου ενός ατόμου και την ανταλλαγή τους με εκείνες ενός άλλου. Για να εκπαιδευτούν μοντέλα βαθιάς μάθησης με σκοπό τη δημιουργία φωτορεαλιστικών εικόνων και βίντεο, απαιτείται συνήθως ένας τεράστιος όγκος δεδομένων. Ως εκ τούτου, το γεγονός ότι οι διασημότητες και οι πολιτικοί έχουν συνήθως πολλές ταινίες και εικόνες διαθέσιμες στο διαδίκτυο τους καθιστά δημοφιλείς στόχους για deepfakes.

Τα GANs είναι αλγόριθμοι βαθιάς μάθησης που προτάθηκαν από τον Ian Goodfellow το 2014, τα οποία εκπαιδεύονται να δημιουργούν ψεύτικες εικόνες, χρησιμοποιώντας έναν γεννήτορα/κωδικοποιητή (encoder) και έναν διακριτή/αποκωδικοποιητή (decoder). [2] Το αρχικό δίκτυο, δηλαδή ο γεννήτορας, δημιουργεί ψεύτικες εικόνες χρησιμοποιώντας το σύνολο δεδομένων εκπαίδευσης. Ο αποκωδικοποιητής στο δεύτερο δίκτυο προσπαθεί να κάνει διάκριση μεταξύ πραγματικών και τεχνητών δημιουργημένων εικόνων. Ο στόχος του γεννήτορα είναι να εξαπατήσει τον διακριτή ώστε να πιστεύει ότι οι δημιουργημένες εικόνες είναι πραγματικές. Κατά τη διάρκεια της εκπαίδευσης, ο γεννήτορας παράγει όλο και πιο ρεαλιστικές φωτογραφίες σε μια προσπάθεια να ξεγελάσει τον διακριτή, ο οποίος συνεχώς βελτιώνεται στον εντοπισμό εικόνων που δημιουργούνται από τεχνητή νοημοσύνη. Μέσω αυτής της διαδικασίας, ένα παραγωγικό αντιπαραθετικό δίκτυο συλλαμβάνει πληροφορίες υψηλού επιπέδου από δεκάδες χιλιάδες φωτογραφίες, προκειμένου να αναπτύξει την ικανότητα αναπαραγωγής παρόμοιων εικόνων στο σύνολο δεδομένων. [6]

Αναλυτικότερα, περιλαμβάνει τρία βασικά βήματα που διευκολύνουν την ακριβή αναπαραγωγή εικόνων. Πρώτον, ο κωδικοποιητής είναι υπεύθυνος για την εξαγωγή κρίσιμων χαρακτηριστικών από την εικόνα εισόδου, συμπιέζοντας την αρχική εικόνα από χιλιάδες εικονοστοιχεία σε εκατοντάδες. Αυτές οι μετρήσεις σχετίζονται με τα χαρακτηριστικά του προσώπου, όπως η κίνηση των ματιών, η στάση του κεφαλιού, ο τόνος του δέρματος και οι συναισθηματικές εκφράσεις. Στη συνέχεια, ο λανθάνων χώρος αντιπροσωπεύει τα μοναδικά χαρακτηριστικά του προσώπου στα οποία εκπαιδεύεται η εικόνα, εστιάζοντας περισσότερο στα κρίσιμα χαρακτηριστικά και αποκλείοντας το θόρυβο και τα ασήμαντα μέρη της εικόνας. Αυτό επιτρέπει την απομνημόνευση των βασικών χαρακτηριστικών του προσώπου ως συμπιεσμένη έκδοση. Τέλος, ο αποκωδικοποιητής αποσυμπιέζει τις πληροφορίες στον λανθάνοντα χώρο για να ανακατασκευάσει την εμφάνιση της αρχικής εικόνας. Η σύγκριση των εικόνων εισόδου και εξόδου παρέχει την απόδοση του αυτόματου κωδικοποιητή, όπου όσο μεγαλύτερη είναι η ομοιότητα της εικόνας εισόδου και εξόδου, τόσο μεγαλύτερη είναι η απόδοση του κωδικοποιητή. Παρά το γεγονός ότι αρχικά επικεντρώθηκε στην όραση του υπολογιστή, η χρήση των GANs γρήγορα επεκτάθηκε και σε άλλους τομείς, συμπεριλαμβανομένου του κειμένου της φυσικής γλώσσας. [1]



Εικόνα 1.2: Διάγραμμα της λειτουργίας των Generative Adversarial Networks (GANs).

Ο Γεννήτορας (Generator) δημιουργεί ψεύτικες εικόνες από δεδομένα εκπαίδευσης και θόρυβο (Noise), ενώ ο Διακριτής (Discriminator) συγκρίνει τις εικόνες και αποφασίζει αν είναι πραγματικές ή ψεύτικες.

1.3 Επιπτώσεις στην Κοινωνία

Η τεχνολογία Deepfake έχει ένα τεράστιο φάσμα εφαρμογών που θα μπορούσαν να χρησιμοποιηθούν τόσο θετικά όσο και αρνητικά, ωστόσο τις περισσότερες φορές εφαρμόζεται για κακόβουλους σκοπούς. Η ανήθικη εφαρμογή της τεχνολογίας Deepfake έχει αρνητικές μακροπρόθεσμες και βραχυπρόθεσμες επιπτώσεις στην κοινωνία μας διότι με μεγάλη ευκολία χειραγωγεί τα συναισθήματα και τις απόψεις των ανθρώπων. Ωστόσο, θα μπορούσαν να υπάρχουν πολλά πλεονεκτήματα με τη σωστή χρήση αυτής της τεχνολογίας. Υπάρχουν διάφοροι λόγοι πίσω από τη δημιουργία περιεχομένου Deepfake, που θα μπορούσε να είναι διασκεδαστικό, αλλά μερικές φορές χρησιμοποιείται για εκδίκηση, εκβιασμό, κλοπή ταυτότητας κάποιου και πολλά άλλα τα οποία θα περιγραφούν λεπτομερώς παρακάτω.

1.3.1 Αρνητικές Επιπτώσεις

Οι άνθρωποι που χρησιμοποιούν τακτικά τα μέσα κοινωνικής δικτύωσης διατρέχουν τεράστιο κίνδυνο απάτης, ειδικά οι πολιτικοί ηγέτες οι οποίοι έχουν μεγάλο αποτύπωμα στο διαδίκτυο. Υπάρχει μεγάλη ανησυχία ότι τα deepfakes θα χρησιμοποιηθούν για να απειλήσουν την εθνική ασφάλεια με τη διάδοση πολιτικής προπαγάνδας. Η τοποθέτηση λέξεων στο στόμα κάποιου σε ένα βίντεο που γίνεται ιογενές είναι ένα ισχυρό όπλο στους σημερινούς πολέμους παραπληροφόρησης, καθώς τέτοια αλλοιωμένα βίντεο μπορούν να διαστρεβλώσουν τη γνώμη των ψηφοφόρων. Η δυνατότητα μαζικής παραγωγής και διάδοσης των DFs από κακόβουλους ανθρώπους μπορεί να αποτελέσει πρόκληση για τη νομιμότητα των διαδικτυακών πολιτικών διαλόγων και να προκαλέσει ζημία σε οποιονδήποτε πολιτικό. Για παράδειγμα, μπορεί να παράγει ένα ψεύτικο βίντεο στο οποίο ένας πολιτικός χρησιμοποιεί φυλετικά επίθετα ή κάνει ρατσιστικές δηλώσεις, έναν προεδρικό υποψήφιο να ομολογεί τη συνενοχή του σε ένα έγκλημα στο οποίο στην πραγματικότητα δεν έχει καμία ανάμειξη. Η παραπληροφόρηση μπορεί να πάρει τεράστιες διαστάσεις και να υποθούν πολύ πιο σοβαρές και ανησυχητικές δηλώσεις, όπως ένας πολιτικός να απειλήσει με πόλεμο κάποιο άλλο έθνος ή να αποκαλύψει ένα κυβερνητικό σχέδιο στο οποίο το στρατιωτικό δυναμικό διαπράττει εγκλήματα πολέμου όπως τη δολοφονία άμαχων πληθυσμών. Σχεδόν όλοι οι παγκόσμιοι ηγέτες, συμπεριλαμβανομένων του Μπαράκ Ομπάμα, πρώην προέδρου των ΗΠΑ, του Ντόναλντ Τραμπ, νυν πρόεδρος των ΗΠΑ, Νάνσι Πελόζι, αμερικανίδας πολιτικού, της Άνγκελα Μέρκελ, γερμανίδας καγκελαρίου, έχουν πέσει θύμα εκμετάλλευσης από ψεύτικα βίντεο που δημιουργήθηκαν με την βοήθεια της βαθιάς μάθησης. [7] Το Deepfake και η τεχνολογία που σχετίζεται με αυτό επεκτείνεται ραγδαία τα τρέχοντα χρόνια και θα υπάρχουν ολοένα και περισσότερα βίντεο ή ακουστικά μέσα με κακόβουλο περιεχόμενο το οποίο θα είναι ακόμα πιο δύσκολο να διακρίνεις το αληθινό από το ψεύτικο. Τέτοια πλαστά βίντεο θα προκαλέσουν αναταραχές, ειδικά την περίοδο των εκλογών, όχι μόνο στους πολίτες αλλά ακόμα και σε αντιστοίχους πολιτικούς ηγέτες άλλων χωρών που θα αισθάνονται απειλή και πιθανόν να επιλέξουν να ενεργήσουν τις εξωτερικές τους πολιτικές οδηγώντας σε διεθνείς συγκρούσεις. [5]

Η χρήση της τεχνολογίας deepfake στην πορνογραφία και τη δημιουργία ψεύτικων βίντεο αποτελεί μία από τις πιο ανησυχητικές αρνητικές επιδράσεις αυτής της τεχνολογίας στην κοινωνία, καθώς πάνω από το 95% των DFs προέρχεται από ταινίες με ερωτικό περιεχόμενο. [8] Το λεγόμενο «deepfake porn», όπου ο όρος αρχικά χρησιμοποιήθηκε αποκλειστικά για αυτήν την πρακτική, αποτελεί έμφυλο φαινόμενο διότι επικεντρώνεται στις γυναίκες. Ειδικότερα, διασημότητες και γυναίκες με μεγάλη επιρροή στα μέσα κοινωνικής δικτύωσης γίνονται συχνά στόχοι DF πορνογραφίας, δεδομένου ότι στο διαδίκτυο υπάρχει πληθώρα υλικού σχετικά με αυτές. Σταρ του Χόλυγουντ, όπως είναι η Scarlett Johansson, η Emma Watson, η Gal Gadot είναι επικρατέστερες διασημότητες που επηρεάζονται από αυτό το φαινόμενο, στις οποίες τα πρόσωπά τους έχουν τοποθετηθεί πάνω από τα πρόσωπα των πορνοστάρ. [1] [5]

Το φαινόμενο του "revenge porn" (εκδικητική πορνογραφία) αποτελεί μία ακόμη σοβαρή διάσταση της χρήσης των DFs. Αυτή η πρακτική δεν περιορίζεται μόνο σε γνωστά πρόσωπα, αλλά επηρεάζει και καθημερινές γυναίκες, καθώς και νεαρές κοπέλες που γίνονται θύματα από πρώην συντρόφους ή ακόμα και από συναδέλφους στον χώρο εργασίας τους. Συνήθως, οι δημιουργοί χρησιμοποιούν προσωπικές εικόνες ή βίντεο χωρίς τη συγκατάθεση των ατόμων, με στόχο την εξευτέλιση και την εκδίκηση. Η ραγδαία εξέλιξη της τεχνολογίας έχει καταστήσει πιο εύκολη από τον οποιοσδήποτε τη δημιουργία κακόβουλου περιεχόμενου, βάζοντας τα άτομα σε παράλογες και επικίνδυνες καταστάσεις. Οι κοινωνικές συνέπειες μπορεί να είναι καταστροφικές για τα θύματα επηρεάζοντας την ψυχική τους υγεία και την προσωπική τους ζωή. Οι γυναίκες που πέφτουν θύματα εκδικητικής πορνογραφίας μπορεί να αντιμετωπίσουν δυσκολίες στην επαγγελματική τους ζωή και να βιώσουν κοινωνική απομόνωση. Επίσης, ένα τέτοιο βίντεο είναι πιθανό να επηρεάσει την εικόνα της εν λόγω γυναίκας, βλέποντας τον εαυτό της να εκτελεί κάθε είδους ρητές ενέργειες μπορεί να έχει αρνητικό αντίκτυπο στην αυτοπεποίθηση και την αυτοεκτίμησή της. Το φαινόμενο των μη συναινετικών βίντεο πορνογραφικού περιεχομένου μπορεί να καταστρέψει τη ζωή οποιασδήποτε γυναίκας, και η ανάγκη για καλύτερη προστασία αυτών των ατόμων είναι πιο σημαντική από ποτέ, ιδιαίτερα τώρα με τη μετεξέλιξη αυτών των βίντεο σε DFs, τα οποία μόνο επιδεινώνουν την κατάσταση. [9]

Τα DFs αναπτύσσονται ολοένα και περισσότερο από απατεώνες με σκοπό τη χειραγώγηση μεγάλων εταιρειών, την κλοπή ταυτότητας, την πλαστογραφία και διάφορα άλλα οικονομικά εγκλήματα. [1] Η δυνατότητα δημιουργίας ρεαλιστικών ψεύτικων βίντεο και ακουστικών μέσων με τη χρήση deepfake τεχνολογίας έχει καταστήσει πιο εύκολη την εξαπάτηση ατόμων και οργανισμών. Για παράδειγμα, θα μπορούσαν να δείχνουν έναν διευθύνοντα σύμβουλο να λέει ρατσιστικές ή μισογυνιστικές προσβολές, να κάνει ψευδείς δηλώσεις πτώχευσης ή να τους απεικονίζει να διαπράττουν έγκλημα. Επιπλέον, η κλοπή ταυτότητας αποτελεί μία άλλη σημαντική απειλή, διότι η υποκείμενη τεχνολογία επιτρέπει την ψηφιακή πλαστοπροσωπία σε πραγματικό χρόνο. Για παράδειγμα, κάποιος να ζητήσει από έναν υπάλληλο να πραγματοποιήσει επείγουσα μεταφορά μετρητών ή να παρέχει εμπιστευτικές πληροφορίες. Αυτό μπορεί να οδηγήσει σε σοβαρές οικονομικές απώλειες εις βάρος της εταιρείας και να απαιτήσει πολύ χρόνο και πόρους για την αποκατάσταση της ζημιάς και την επαναφορά της πραγματικής τους ταυτότητας. Τέλος, η παραποίηση ταυτότητας, διαβατηρίου και άλλων εγγράφων τα οποία μπορούν να παραποιηθούν με εντυπωσιακή ακρίβεια, διευκολύνουν την είσοδο εγκληματιών σε προστατευόμενους χώρους και τη διάπραξη οικονομικών εγκλημάτων. [5] Η απάτη και η κλοπή

ταυτότητας έχουν διαπραχθεί εδώ και αιώνες και τα DF αποτελούν σοβαρή απειλή για την ασφάλεια των ατόμων και των επιχειρήσεων. [9]

Τα deepfakes είναι πιθανό να παρεμποδίσουν την ψηφιακή παιδεία και την εμπιστοσύνη των πολιτών στις πληροφορίες που παρέχονται από αξιόπιστες πηγές. Η δημιουργία ψεύτικων βίντεο που δείχνουν κυβερνητικούς αξιωματούχους να εκφράζουν δηλώσεις ή γεγονότα που δεν συνέβησαν ποτέ, μπορεί να προκαλέσει αμφιβολίες στους ανθρώπους σχετικά με την εγκυρότητα και την ακρίβεια των πληροφοριών. Οι άνθρωποι σήμερα επηρεάζονται όλο και περισσότερο από ανεπιθύμητα μηνύματα που δημιουργούνται από τεχνητή νοημοσύνη και από ψεύτικες ειδήσεις που βασίζονται σε πλαστό περιεχόμενο, όπως πλαστά βίντεο και πληθώρα θεωριών συνωμοσίας. Ωστόσο, η πιο επιζήμια πτυχή των deepfakes μπορεί να μην είναι η παραπληροφόρηση καθαυτή, αλλά το πώς η συνεχής επαφή με την παραπληροφόρηση οδηγεί τους ανθρώπους να αμφισβητούν την εγκυρότητα των πληροφοριών που λαμβάνουν. [10] Αυτή η ανασφάλεια μπορεί να οδηγήσει σε ένα φαινόμενο που ονομάζεται "απάθεια πραγματικότητας", όπου οι άνθρωποι αρχίζουν να αμφισβητούν την αξιοπιστία οποιασδήποτε πληροφορίας, ακόμα και αν είναι αληθής. Με άλλα λόγια, η μεγαλύτερη απειλή δεν είναι ότι οι άνθρωποι θα εξαπατηθούν, αλλά ότι θα θεωρήσουν τα πάντα ως εξαπάτηση. [5]

1.3.2 Θετικές Επιπτώσεις

Όπως αναλύσαμε και παραπάνω, η τεχνολογία DeepFake συνδέεται συχνά με αρνητικές επιδράσεις, ωστόσο εξακολουθεί να έχει κάποιες θετικές χρήσεις στην κοινωνία σε διάφορους τομείς. Ένας από αυτούς είναι η κινηματογραφική βιομηχανία, η οποία επωφελείται από την τεχνολογία DF με πολλούς τρόπους. Για παράδειγμα, οι δημιουργοί ταινιών μπορούν να αναδημιουργήσουν ψηφιακά ηθοποιούς που δεν είναι διαθέσιμοι, είτε λόγω θανάτου είτε λόγω άλλων περιορισμών ή να χρησιμοποιήσουν ειδικά εφέ και προηγμένη επεξεργασία προσώπου, βελτιώνοντας την ποιότητα των οπτικών αφηγήσεων. Επιπλέον, η τεχνολογία DF είναι ικανή να δημιουργήσει ψηφιακές φωνές για ηθοποιούς που έχασαν τις δικές τους ή για την ενημέρωση του κινηματογραφικού υλικού αντί για την ανακατασκευή τους. Η τεχνολογία deepfake επιτρέπει επίσης αυτόματη και ρεαλιστική φωνητική μεταγλώττιση για ταινίες σε οποιαδήποτε επιθυμητή γλώσσα. Γίνεται αντιληπτό, ότι μπορεί να εξοικονομηθεί τεράστιο ποσό χρημάτων αλλά και χρόνου της κινηματογραφικής βιομηχανίας, χρησιμοποιώντας τις δυνατότητες που σου προσφέρει το DF. Μια παγκόσμια εκστρατεία ευαισθητοποίησης για την ελονοσία του 2019 με τον David Beckham έσπασε τα γλωσσικά εμπόδια μέσω μιας εκπαιδευτικής διαφήμισης που χρησιμοποίησε οπτική και φωνητική τεχνολογία για να τον κάνει να φαίνεται πολύγλωσσος. [5]

Οι επιχειρήσεις ενδιαφέρονται για τις δυνατότητες της τεχνολογίας deepfake, καθώς μπορεί να μεταμορφώσει το ηλεκτρονικό εμπόριο και τη διαφήμιση με σημαντικούς τρόπους. Τα deepfakes επιτρέπουν τη δημιουργία εξατομικευμένου περιεχομένου, δηλαδή, βοηθάει τους καταναλωτές να δουν πώς θα φαίνονται τα προϊόντα πάνω τους, χρησιμοποιώντας βίντεο ή εικόνες των ίδιων, δημιουργώντας μια πιο προσωπική εμπειρία αγοράς. Η τεχνολογία επιτρέπει την εικονική προσαρμογή για την προεπισκόπηση του τρόπου εμφάνισης ενός ενδύματος πριν από την αγορά και μπορεί να δημιουργήσει στοχευμένες διαφημίσεις μόδας. Μία από τις εφαρμογές αυτής της τεχνολογίας είναι η δυνατότητα γρήγορης δοκιμής ρούχων στο διαδίκτυο. Η τεχνολογία όχι μόνο επιτρέπει στους ανθρώπους να δημιουργούν ψηφιακούς κλώνους του εαυτού τους αλλά και να έχουν αυτά τα προσωπικά είδωλα να «ταξιδεύουν» μαζί τους σε ηλεκτρονικά καταστήματα. [5]

Η τεχνολογία Deepfake έχει προσφέρει εξαιρετικές δυνατότητες στην εκπαίδευση, ειδικότερα στον τομέα της ιστορικής αναπαράστασης. Η χρήση της τεχνολογίας αυτής επιτρέπει τη δημιουργία ρεαλιστικών αναπαραστάσεων ιστορικών προσώπων, καθιστώντας την εκπαιδευτική διαδικασία πιο άμεση και βιωματική. Πρόσφατα, το Μουσείο Dali που βρίσκεται στην Φλόριντα έδωσε την ευκαιρία στους επισκέπτες της να συναντήσουν τον Salvador Dali και να ασχοληθούν με τη ζωή του πιο διαδραστικά, για να γνωρίσουν αυτή τη μεγάλη προσωπικότητα μέσω της τεχνητής νοημοσύνης. Αυτό όχι μόνο ενισχύει την εμπειρία της μάθησης, αλλά κάνει την ιστορία πιο προσιτή και διασκεδαστική προς τους μαθητές. [7]

1.4 Τρόποι Ανίχνευσης

Είναι συχνά δύσκολο και μερικές φορές αδύνατο να εντοπιστεί deepfake περιεχόμενο από έναν άνθρωπο με ανεκπαίδευτα μάτια. Σταδιακά γίνεται όλο και πιο απλή και αποτελεσματική η δημιουργία ενός deepfake, γι' αυτόν το λόγο απαιτείται ένα καλό επίπεδο εξειδίκευσης για τον εντοπισμό παρατυπιών. Λόγω αυτού και των προαναφερθέντων κινδύνων, καθίσταται επιτακτική η ανάπτυξη μεθόδων ανίχνευσης αυτών των ψεύτικων βίντεο. Μέχρι τώρα έχουν προταθεί αρκετές προσεγγίσεις, συμπεριλαμβανομένης της ανίχνευσης μηχανών, της εγκληματολογίας, της εξακρίβωσης της ταυτότητας καθώς και της ρύθμισης για την καταπολέμηση του deepfake. [7]

Όταν πρωτοεμφανίστηκαν τα deepfakes ήταν πιο εύκολο για κάποιον να τα ανιχνεύσει με γυμνό μάτι παρατηρώντας ορισμένα συγκεκριμένα σημεία που συχνά παρουσιάζουν ανωμαλίες. Πρώτον, οι ασυνέπειες στις κινήσεις του στόματος και ο αταίριαστος συγχρονισμός μεταξύ ήχου και χειλιών μπορεί να αποκαλύψουν μια πλαστή δημιουργία. Επίσης, τα μάτια στα deepfake βίντεο ενδέχεται να μην ανοιγοκλείνουν φυσικά ή να παραμένουν αφύσικα ανοιχτά. Η χαμηλή ποιότητα ή οι ανωμαλίες στις σκιές και τα αντανακλαστικά στο πρόσωπο ή στα μαλλιά μπορεί επίσης να είναι ενδεικτικές. Επιπλέον, παρατηρώντας λεπτομέρειες όπως ασυμμετρίες στο πρόσωπο, αφύσικα ομαλή επιδερμίδα, ή μικρές αλλοιώσεις στα άκρα των ματιών και του στόματος μπορεί κανείς να αντιληφθεί την τεχνητή προέλευση του περιεχομένου. [11]

Με την εξέλιξη της τεχνολογίας, αυτές οι λεπτομέρειες γίνονται ακόμα πιο δύσκολα αντιληπτές από τον άνθρωπο, καθώς τα βίντεο τείνουν να φαίνονται ολοένα και πιο ρεαλιστικά. Ωστόσο, το πρόβλημα δεν περιορίζεται σε αυτό, καθώς δεν αυξάνεται μόνο η ποιότητα των βίντεο αλλά και ο αριθμός αυτών. Πλέον, υπάρχουν χιλιάδες βίντεο διασκορπισμένα στο διαδίκτυο, καθιστώντας ανέφικτο να έχουμε μόνο ειδικούς να πραγματοποιούν την ανίχνευση. Γι' αυτόν το λόγο, είναι πιο αναγκαίο από ποτέ να αναπτυχθούν μέθοδοι ανίχνευσης με αλγορίθμους βαθιάς μάθησης, οι οποίοι θα διευκολύνουν τη διαδικασία και θα την καταστήσουν όσο το δυνατό πιο αποδοτική. Αυτοί οι αλγόριθμοι μπορούν να αναλύουν τα δεδομένα με ακρίβεια και ταχύτητα, προσφέροντας μια αποτελεσματική λύση στην αντιμετώπιση της διάδοσης πλαστών βίντεο και προστατεύοντας την αυθεντικότητα της πληροφορίας. [6]

1.5 Σχετική Έρευνα

Οι M. M. El-Gayar και οι συνεργάτες του χρησιμοποιούν μια προηγμένη μέθοδο για ανίχνευση deepfake βίντεο που αξιοποιεί τα νευρωνικά δίκτυα γραφημάτων (GNNs) και τα συνελικτικά νευρωνικά δίκτυα (CNNs). Η μέθοδος αυτή συνδυάζει τη δυναμική προσαρμογή των γειτονικών κόμβων του GNN με την ικανότητα ανίχνευσης χαρακτηριστικών του CNN. Το μοντέλο χρησιμοποιεί μια ιεραρχική δομή για την καταγραφή λεπτών χαρακτηριστικών, βελτιώνοντας την αναπαράσταση και την ακρίβεια ανίχνευσης. Η διαδικασία ανίχνευσης περιλαμβάνει δύο φάσεις: μια ροή μίνι- παρτίδας (mini-batch) και μια ροή CNN τεσσάρων μπλοκ, οι οποίες ενσωματώνονται μέσω τριών δικτύων σύντηξης (FuNet-a, FuNet-M, FuNet-C). Η ακρίβεια της μεθόδου σε διαφορετικά σύνολα δεδομένων ήταν εξαιρετική, με ποσοστά 95,09% στο σύνολο δεδομένων FF++, 99,3% στο DFDC και 98,9% στο Celeb-DF, αποδεικνύοντας την υψηλή αποδοτικότητα και αξιοπιστία της προσέγγισης αυτής. [4]

Μια άλλη μέθοδος που προτάθηκε από τους Zeina Ayman και τους συνεργάτες του χρησιμοποιεί το μοντέλο VGG (Visual Geometry Group) για την ανίχνευση deepfake βίντεο. Το VGG είναι ένα προεκπαιδευμένο νευρωνικό δίκτυο βασισμένο στην αρχιτεκτονική CNN με πέντε μπλοκ συνελίξεων και max pooling, το οποίο κατέγραψε ακρίβεια περίπου 88%. Αντίθετα, το CNN περιλαμβάνει την προεπεξεργασία των δεδομένων με εξαγωγή προσώπων από τα βίντεο και αναπροσαρμογή τους σε διαστάσεις 224x224 pixels. Το CNN, με την ικανότητά του να ανιχνεύει πρότυπα και λεπτομέρειες στις εικόνες, πέτυχε ακρίβεια περίπου 94%. Αυτά τα αποτελέσματα υποδεικνύουν ότι το CNN υπερτερεί του VGG στην ανίχνευση deepfake. [12]

Ένα άλλο παράδειγμα από τους Haliassos και τους συνεργάτες του παρουσιάζει μία καινοτόμο προσέγγιση για την ανίχνευση deepfake βίντεο, βασισμένη στην αυτο-επιτήρηση (self-supervision). Η μέθοδος ονομάζεται RealForensics και αποτελείται από δύο στάδια. Το πρώτο στάδιο περιλαμβάνει την εκμάθηση χρονικά πυκνών αναπαραστάσεων βίντεο, χρησιμοποιώντας διατροπική αυτο-επιτήρηση από πολλά φυσικά ομιλούντα πρόσωπα. Αυτές οι αναπαραστάσεις χρησιμοποιούνται ως στόχοι πρόβλεψης στο δεύτερο στάδιο για την κανονικοποίηση της δυαδικής ταξινόμησης πλαστογραφίας (classification). Η μέθοδος πέτυχε ακρίβεια 97.1% και 95.7% στα σύνολα δεδομένων FaceShifter και DeeperForensics αντίστοιχα. [13]

Τέλος, οι Feng και οι συνεργάτες του αναπτύσσουν μία μέθοδο που βασίζεται στην ανίχνευση ανωμαλιών μεταξύ του ήχου και της εικόνας σε βίντεο. Χρησιμοποιούν ένα πολυτροπικό μοντέλο, το οποίο συνδυάζει πληροφορίες από το οπτικό και το ακουστικό κομμάτι του βίντεο. Το μοντέλο εκπαιδεύεται με πραγματικά δεδομένα για την ανίχνευση ανωμαλιών, χωρίς να απαιτείται εκπαίδευση σε δεδομένα deepfake. Με βάση αυτές τις ανωμαλίες το σύστημα προσδιορίζει εάν το βίντεο είναι πλαστό. Η μέθοδος επιτυγχάνει υψηλή ακρίβεια, με επιδόσεις στο 97,4% σε δοκιμές σε κοινά σύνολα δεδομένων deepfake, όπως το FaceForensics++. [14]

1.6 Προκλήσεις και Περιορισμοί στην Ανίχνευση και Αντιμετώπιση

Το πρόβλημα με τα deepfakes δεν αφορά μόνο την απόδειξη ότι κάτι είναι ψευδές, αλλά και την απόδειξη ότι ένα αντικείμενο είναι αυθεντικό. Η ανίχνευση και η αντιμετώπιση των deepfakes παρουσιάζουν πολυάριθμους περιορισμούς που καθιστούν τη διαδικασία ιδιαίτερα περίπλοκη. Αυτές οι προκλήσεις αφορούν τόσο τεχνικούς όσο και κοινωνικούς παράγοντες, επηρεάζοντας την αποτελεσματικότητα και την αποδοτικότητα των μεθόδων ανίχνευσης.

1.6.1 Κοινωνικές και Νομικές Προκλήσεις

Η ευαισθητοποίηση του κοινού σχετικά με τους κινδύνους και τις συνέπειες του deepfake είναι περιορισμένη. Παρά την εκτενή κάλυψη από τα μέσα ενημέρωσης και τις ανησυχίες που εκφράζουν οι αρχές, η απειλή των deepfakes δεν έχει γίνει πλήρως αντιληπτή από το κοινό. Γενικά, υπάρχει ανάγκη για ενημέρωση του κοινού σχετικά με τις δυνατότητες κατάχρησης της τεχνητής νοημοσύνης. Κατά τη διάρκεια της εκπαιδευτικής διδασκαλίας συνιστάται η εκμάθηση της κριτικής σκέψης και της ψηφιακής παιδείας, καθώς αυτά τα χαρακτηριστικά συμβάλλουν στην ικανότητα των παιδιών να εντοπίζουν ψεύτικες ειδήσεις και να αλληλοεπιδρούν με μεγαλύτερο σεβασμό μεταξύ τους στο Διαδίκτυο. Αυτές οι δεξιότητες θα πρέπει να προωθηθούν μεταξύ του μεγαλύτερου, λιγότερο εξοικειωμένου με την τεχνολογία πληθυσμού. Είναι απαραίτητο οι άνθρωποι να είναι σε θέση να αξιολογήσουν κριτικά την αυθεντικότητα και το κοινωνικό πλαίσιο ενός βίντεο που μπορεί να επιθυμούν να καταναλώσουν, καθώς και την αξιοπιστία της πηγής του. Είναι επίσης σημαντικό να θυμόμαστε ότι η ποιότητα δεν αποτελεί ένδειξη της αυθεντικότητας ενός βίντεο, καθώς τα σύγχρονα deepfakes μπορούν να δημιουργηθούν με τόσο υψηλή ποιότητα που μπορούν εύκολα να παραπλανήσουν ακόμα και τους πιο έμπειρους χρήστες. [5]

Η αυθεντικότητα του βίντεο είναι ιδιαίτερα σημαντική για τις εταιρείες μέσων ενημέρωσης που πρέπει να διασφαλίζουν την αξιοπιστία του περιεχομένου τους. Σε ένα περιβάλλον όπου οι πληροφορίες διαχέονται ταχύτατα και συχνά χωρίς έλεγχο, η ταυτοποίηση του δημιουργού, της προέλευσης και της διανομής του βίντεο μπορεί να αποδειχθεί δύσκολη. Η συνεργασία με πλατφόρμες κοινωνικών μέσων και η ανάπτυξη εργαλείων για την ταχεία ανίχνευση και αφαίρεση των deepfakes είναι κρίσιμη για την αντιμετώπιση αυτής της πρόκλησης. Παρόλα αυτά, λίγες εταιρείες κοινωνικών μέσων έχουν ακόμη πολιτικές σχετικά με τα deepfakes. Γι’ αυτό το λόγο, η συνεργασία μεταξύ τους είναι απαραίτητη για να αποτραπεί η κατάχρηση των πλατφορμών τους για παραπληροφόρηση. Ενδεικτικά, πλατφόρμες όπως το Reddit έχουν απαγορεύσει τη διακίνηση deepfake πορνογραφίας και άλλου μη συναινετικού υλικού, επισημαίνοντας τη σημασία της ενεργής εμπλοκής των χρηστών για την ανίχνευση και την αφαίρεση τέτοιου περιεχομένου. Επιπλέον, το Facebook αποτρέπει την προβολή διαφημίσεων και την παραγωγή εσόδων από κάθε περιεχόμενο που αναγνωρίζεται ως ψευδές ή παραπλανητικό. [5]

Επί του παρόντος, τα deepfakes δεν αντιμετωπίζονται ειδικά από αστικούς ή ποινικούς νόμους. Ωστόσο, νομικοί εμπειρογνώμονες έχουν προτείνει την προσαρμογή των υφιστάμενων νομικών πλαισίων για να καλύψουν ζητήματα όπως είναι η δυσφήμιση, η απάτη ταυτότητας ή η πλαστοπροσωπία χρησιμοποιώντας deepfakes. Παρά τις προτάσεις αυτές, οι υπάρχουσες νομικές διατάξεις δεν επαρκούν για να αντιμετωπίσουν όλες τις περιπτώσεις παραποίησης, καθώς η δημιουργία deepfake μπορεί να παραβιάζει δικαιώματα προσωπικότητας και πνευματικής ιδιοκτησίας. Οι εφαρμογές της τεχνολογίας αυτής εξελίσσονται ταχύτερα από την ικανότητα των νομοθετικών συστημάτων να αντιδράσουν. Επίσης, η δημοκρατικοποίηση της τεχνολογίας σημαίνει ότι όλο και περισσότεροι πολίτες μπορούν να δημιουργήσουν deepfakes, γεγονός που καθιστά την εφαρμογή των νόμων και την επιβολή ποινών ιδιαίτερα δύσκολη. Έτσι, οι ρυθμιστικές αρχές πρέπει να περιηγηθούν σε ένα δύσκολο νομικό πλαίσιο που περιλαμβάνει ζητήματα ελευθερίας του λόγου, προκειμένου να ρυθμίσουν σωστά τη χρήση της τεχνολογίας deepfake. [9]

1.6.2 Τεχνικές Προκλήσεις

Η απόδοση ενός μοντέλου ανίχνευσης deepfake εξαρτάται από την ποικιλία των μεγάλων συνόλων δεδομένων που χρησιμοποιούνται κατά τη διάρκεια της εκπαίδευσής του. Ωστόσο, η έλλειψη ποιοτικών δεδομένων παραμένει πρόκληση. Όταν τα μοντέλα δοκιμάζονται σε περιεχόμενο με άγνωστο τύπο χειραγώγησης, τότε η ικανότητα εντοπισμού αυτών των παρατυπιών γίνεται δύσκολη. Οι λειτουργίες μετα-επεξεργασίας που εφαρμόζονται σε πολυμέσα deepfake όπως είναι η αφαίρεση χρονικών τεχνημάτων, η θόλωση, η εξομάλυνση και η περικοπή χρησιμοποιούνται συχνά με σκοπό να παραπλανήσουν τον ανιχνευτή, καθιστώντας ακόμα πιο περίπλοκη την ανίχνευση. [2] Επιπλέον, τα υπάρχοντα σύνολα δεδομένων συχνά περιέχουν βίντεο χαμηλής ποιότητας, με ορατές ατέλειες όπως χαμηλής ποιότητας συνθετικών προσώπων, ορατά όρια σύνδεσης και αναντιστοιχία χρωμάτων. Αυτά τα αντικείμενα είναι πιθανώς το αποτέλεσμα ατελών βημάτων της μεθόδου σύνθεσης και της έλλειψης επιμέλειας των συνθετικών βίντεο πριν συμπεριληφθούν στα σύνολα δεδομένων. Τα βίντεο deepfake με τόσο χαμηλές οπτικές ιδιότητες δύσκολα μπορούν να είναι πειστικά και δυσχεραίνουν την ανάπτυξη αξιόπιστων μεθόδων ανίχνευσης.

Ένα ακόμη σημαντικό ζήτημα είναι ότι οι μέθοδοι ανίχνευσης deepfake που εκπαιδεύονται χρησιμοποιώντας συγκεκριμένα σύνολα δεδομένων έχουν πρόβλημα να διατηρήσουν την απόδοση σε διαφορετικά σύνολα δεδομένων. Αυτή η έλλειψη γενίκευσης σημαίνει ότι τα μοντέλα μπορεί να αποτύχουν όταν έρθουν αντιμέτωπα με νέους ή διαφορετικούς τύπους χειραγώγησης που δεν είχαν συμπεριληφθεί στο αρχικό σύνολο εκπαίδευσης. Η συνεχής ενημέρωση και επέκταση των συνόλων δεδομένων είναι αναγκαία για να διασφαλιστεί ότι τα μοντέλα ανίχνευσης παραμένουν αποτελεσματικά απέναντι στις διαρκώς εξελισσόμενες τεχνικές δημιουργίας deepfake. [3]

Η χρήση βοηθητικών συνόλων δεδομένων για την εκπαίδευση ανίχνευσης deepfake βελτιώνει σημαντικά την ακρίβεια και την αποτελεσματικότητα αυτών των μοντέλων, ωστόσο αυτή η προσέγγιση συνοδεύεται από αυξημένο κόστος υψηλότερων υπολογιστικών απαιτήσεων. Η ανάγκη για επεξεργασία και ανάλυση μεγάλων και ποικιλόμορφων συνόλων δεδομένων απαιτεί μεγαλύτερη υπολογιστική ισχύ και μνήμη, καθώς και περισσότερο χρόνο εκπαίδευσης. Αυτός ο παράγοντας αποτελεί σημαντικό περιορισμό, ιδίως για οργανισμούς που δεν διαθέτουν τους απαραίτητους πόρους. [13]

Συνήθως, τα μοντέλα ανίχνευσης deepfake εκπαιδεύονται με μεγάλα σύνολα δεδομένων που περιλαμβάνουν ποικίλες περιπτώσεις πλαστογραφίας. Ωστόσο, σε ορισμένες περιπτώσεις, όπως στη δημοσιογραφία ή στην επιβολή του νόμου με βάση την ανίχνευση deepfake, μπορεί να είναι διαθέσιμο μόνο ένα μικρό σύνολο δεδομένων. Επιπλέον, αυτό το είδος συνόλου δεδομένων απαιτεί πρόσθετη προσπάθεια για την επισήμανση και την ταξινόμηση ώστε να είναι ξεκάθαρο ποιος τύπος πλαστογραφίας χρησιμοποιείται. Αυτή η διαδικασία μπορεί να είναι χρονοβόρα και απαιτητική σε πόρους, καθώς χρειάζεται εξειδικευμένη γνώση για την αναγνώριση και την κατηγοριοποίηση των διαφορετικών τεχνικών παραποίησης. Απαιτείται περαιτέρω μελέτη και ανάλυση για την καλύτερη κατανόηση της δημοσιογραφίας και των αρχών επιβολής του νόμου, ώστε να αναπτυχθούν προσαρμοσμένες λύσεις που να ανταποκρίνονται στις ιδιαίτερες προκλήσεις αυτών των τομέων. [2]

2. Θεωρητικό Υπόβαθρο

Η τεχνητή νοημοσύνη (AI) αποτελεί έναν από τους κύριους πυλώνες της σύγχρονης τεχνολογίας, διαμορφώνοντας ριζικά το τοπίο σε πολλούς τομείς. Ένα από τα πιο εντυπωσιακά και καθοριστικά επιτεύγματά της είναι τα συνελικτικά νευρωνικά δίκτυα (ΣΝΔ, CNNs), τα οποία ανήκουν στην κατηγορία των βαθιών νευρωνικών δικτύων. Τα συνελικτικά νευρωνικά δίκτυα έχουν τη δυνατότητα να μαθαίνουν αυτόματα μια ιεραρχία χαρακτηριστικών από τα δεδομένα, ξεκινώντας από απλά μοτίβα, όπως γραμμές και άκρες, και προχωρώντας σε πιο σύνθετες δομές, όπως σχήματα και αντικείμενα. Αυτή η προσέγγιση υπερέχει έναντι της παραδοσιακής μεθόδου, όπου τα χαρακτηριστικά καθορίζονται χειροκίνητα από ειδικούς, καθώς επιτρέπει στα ΣΝΔ να προσαρμόζονται καλύτερα σε κάθε πρόβλημα και να βελτιώνουν την ακρίβεια στην ταξινόμηση (classification). Η αυτόματη εκμάθηση χαρακτηριστικών καθιστά τα ΣΝΔ εξαιρετικά αποτελεσματικά σε εφαρμογές όπως η αναγνώριση εικόνας, η επεξεργασία φυσικής γλώσσας (Natural language processing, NLP) και η ανίχνευση αντικειμένων. [15]

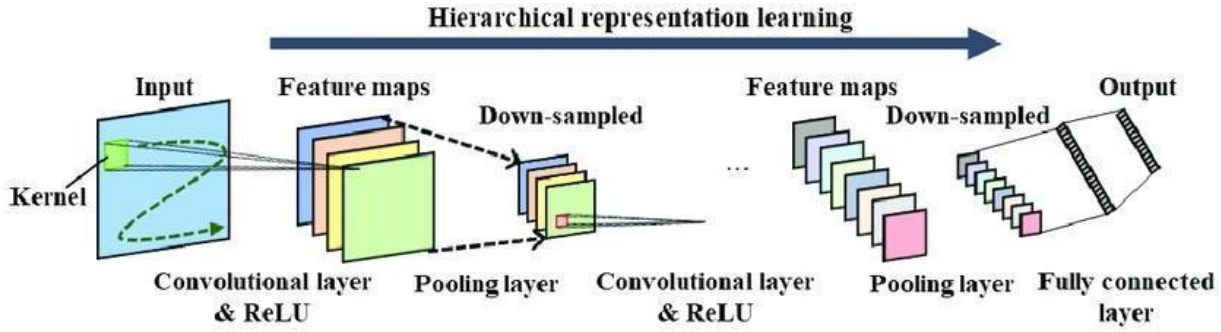
Τα συνελικτικά νευρωνικά δίκτυα πήραν το όνομά τους από τη μαθηματική πράξη της συνέλιξης (convolution), η οποία συνδυάζει διάφορες μαθηματικές λειτουργίες. Η πρώτη αρχιτεκτονική προτάθηκε από τον LeCun και τους συνεργάτες του το 1990 για την αναγνώριση χειρόγραφων ψηφίων. Η πραγματική πρόοδος, ωστόσο, σημειώθηκε το 2012 με το AlexNet, ένα ΣΝΔ που ανέπτυξε ο Krizhevsky και οι συνεργάτες του, μειώνοντας σημαντικά το ποσοστό σφάλματος στην αναγνώριση εικόνων. Έκτοτε, τα ΣΝΔ έχουν εξελιχθεί σε βασικό εργαλείο για πλήθος εφαρμογών, όπως η αναγνώριση αντικειμένων και η ανάλυση εικόνας. [16]

Ακριβώς επειδή η τεχνολογία deepfake και η ανίχνευσή τους βασίζονται σε τεχνικές αναγνώρισης βίντεο και εικόνας, τα συνελικτικά νευρωνικά δίκτυα αποτελούν σημαντική βάση για την επεξεργασία αυτών των δεδομένων. Χρησιμοποιώντας την ικανότητά τους να μαθαίνουν από μεγάλες ποσότητες δεδομένων, τα ΣΝΔ διευκολύνουν την ανάπτυξη μεθόδων ανίχνευσης των λεγόμενων deepfakes, επιτρέποντας έτσι την αναγνώριση και την αποτροπή κακόβουλων χρήσεων αυτής της τεχνολογίας.

2.1 Δομή Συνελικτικών Νευρωνικών Δικτύων

Τα συνελικτικά νευρωνικά δίκτυα έχουν πολλαπλά επίπεδα ιεραρχικής δομής που τα καθιστούν εξαιρετικά αποδοτικά στην επεξεργασία εικόνας και βίντεο. Τα ΣΝΔ αποτελούνται από κρυφά επίπεδα, τα οποία δεν είναι πλήρως συνδεδεμένα με τα προηγούμενα, γεγονός που μειώνει τον αριθμό των παραμέτρων και βελτιώνει την αποδοτικότητα του δικτύου. Σε κάθε επίπεδο, όπως και άλλα τεχνητά νευρωνικά δίκτυα, υπάρχουν νευρώνες με μαθησιακά βάρη και πολώσεις (biases) που προσαρμόζονται κατά την εκπαίδευση. Παρόλο που κάθε νευρώνας δεν συνδέεται άμεσα με όλους τους νευρώνες του προηγούμενου επιπέδου, λαμβάνει πολλαπλές εισροές, υπολογίζει το σταθμισμένο άθροισμα των εισόδων και των βαρών, το εφαρμόζει σε μια συνάρτηση ενεργοποίησης και παράγει την επιθυμητή έξοδο. [15]

Η βασική δομή ενός ΣΝΔ περιλαμβάνει τα παρακάτω κύρια επίπεδα: το συνελικτικό επίπεδο (convolution layer), το επίπεδο υποδειγματοληψίας (pooling layer) και το πλήρως συνδεδεμένο επίπεδο (fully connected layer). Σε κάθε επίπεδο, οι νευρώνες λαμβάνουν πολλαπλές εισόδους, υπολογίζουν το σταθμισμένο άθροισμά τους και εφαρμόζουν μια συνάρτηση ενεργοποίησης για να παραχθεί η επιθυμητή έξοδος. Χάρη στην ικανότητά τους να μαθαίνουν αυτόματα από τα δεδομένα και να προσαρμόζονται σε διαφορετικές δομές, τα ΣΝΔ χρησιμοποιούνται ευρέως σε εφαρμογές όπως η αναγνώριση εικόνας και η επεξεργασία φυσικής γλώσσας. [15]



Εικόνα 2.1: Γενική δομή ενός συνελεκτικού νευρωνικού δικτύου.

Η εικόνα απεικονίζει την ιεραρχική διαδικασία εκμάθησης χαρακτηριστικών. Αρχικά, η είσοδος (input) αποτελείται από τα αρχικά δεδομένα που θα επεξεργαστεί το δίκτυο. Ένας μικρός πυρήνας (kernel) «σαρώνει» την είσοδο και εξάγει συγκεκριμένα χαρακτηριστικά, δημιουργώντας χάρτες χαρακτηριστικών (feature Maps). Το πρώτο συνελικτικό επίπεδο (convolutional layer) εφαρμόζει τη συνέλιξη (convolution) σε συνδυασμό με τη συνάρτηση ενεργοποίησης ReLU, παράγοντας χάρτες χαρακτηριστικών. Στη συνέχεια, το συγκεντρωτικό επίπεδο (pooling layer) μειώνει τις χωρικές διαστάσεις αυτών των χαρτών (down-sampling). Αυτή η διαδικασία επαναλαμβάνεται σε διαδοχικά επίπεδα συνελίξεων και συγκεντρώσεων. Στο τέλος, το πλήρως συνδεδεμένο επίπεδο (fully connected layer) μετατρέπει τους χάρτες χαρακτηριστικών σε ένα επίπεδο μονοδιάστατο διάνυσμα και παράγει την τελική έξοδο.

2.2 Συνελικτικό Επίπεδο (Convolution Layer)

Το συνελικτικό επίπεδο είναι το βασικό δομικό στοιχείο ενός ΣΝΔ, καθώς εκτελεί την πράξη της συνέλιξης στην είσοδο με τη χρήση φίλτρων. Τα φίλτρα αυτά, τα οποία είναι πίνακες μικρού μεγέθους, μετακινούνται πάνω στην είσοδο και υπολογίζουν το εσωτερικό γινόμενο μεταξύ του φίλτρου και μιας μικρής περιοχής της εισόδου σε κάθε βήμα. Το αποτέλεσμα αυτής της συνέλιξης είναι ένας χάρτης χαρακτηριστικών (feature maps), ο οποίος συγκεντρώνει την πληροφορία της εισόδου σε μία μόνο τιμή, μειώνοντας το μέγεθος των δεδομένων και εξάγοντας χρήσιμα χαρακτηριστικά για το επόμενο επίπεδο. [16]

2.2.1 Συνέλιξη (Convolution)

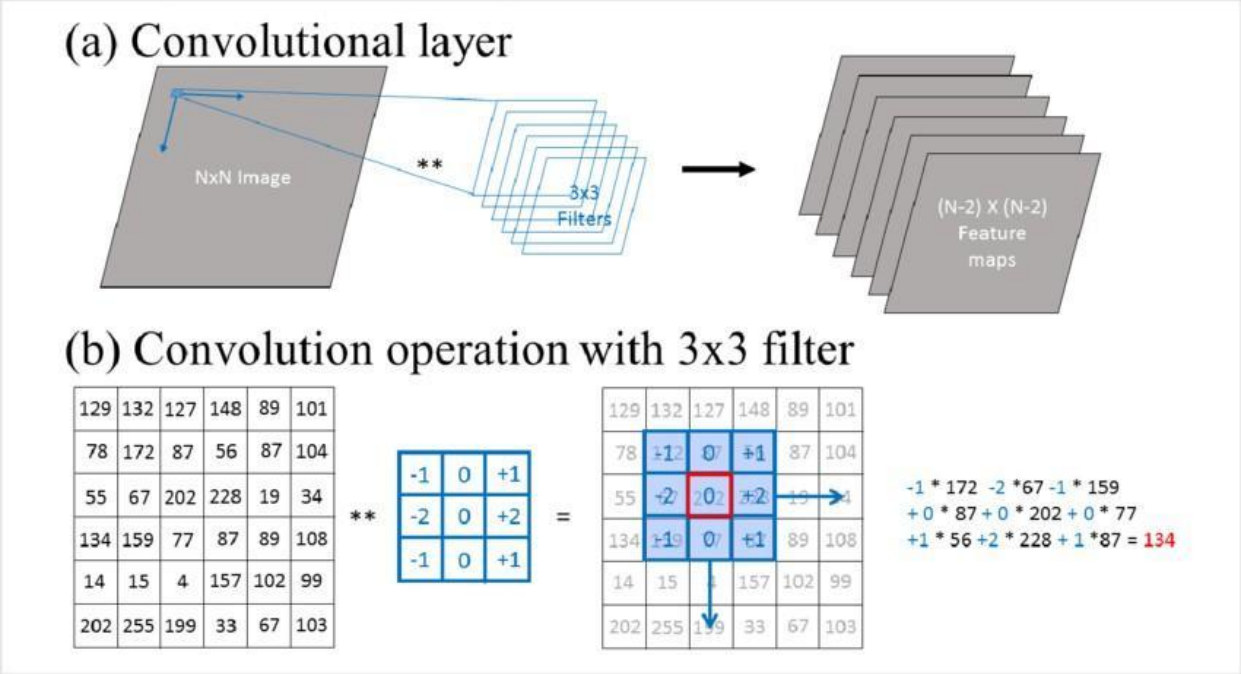
Η συνέλιξη (convolution) είναι μια μαθηματική πράξη που χρησιμοποιείται ευρέως στην επεξεργασία σήματος, εικόνας και στην υπολογιστική όραση, και αποτελεί θεμελιώδη λειτουργία των ΣΝΔ. Η συνέλιξη χρησιμοποιείται για την εξαγωγή χαρακτηριστικών από εικόνες, μετακινώντας ένα φίλτρο ή πυρήνα πάνω στην εικόνα εισόδου και υπολογίζοντας το εσωτερικό γινόμενο σε κάθε θέση. Η μαθηματική σχέση για τη συνέλιξη δύο διακριτών συναρτήσεων f και g δίνεται από τον τύπο:

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m] \cdot g[n-m]$$

Όπου:

- f είναι η συνάρτηση εισόδου,
- g είναι το φίλτρο ή πυρήνας,
- n είναι ο δείκτης της θέσης της εξόδου,
- m αντιστοιχεί σε μια θέση μέσα στην περιοχή της εικόνας.

Οι παράμετροι όπως το μέγεθος του φίλτρου, το βήμα (stride) και η προσθήκη μηδενικών γύρω από την εικόνα (zero padding), καθορίζουν το μέγεθος και την ανάλυση του παραγόμενου χάρτη χαρακτηριστικών. Η συνέλιξη επιτρέπει την απομόνωση σημαντικών μοτίβων και τη μείωση των διαστάσεων των δεδομένων, καθιστώντας τα ΣΝΔ αποτελεσματικά στην επεξεργασία και αναγνώριση εικόνων. [17]



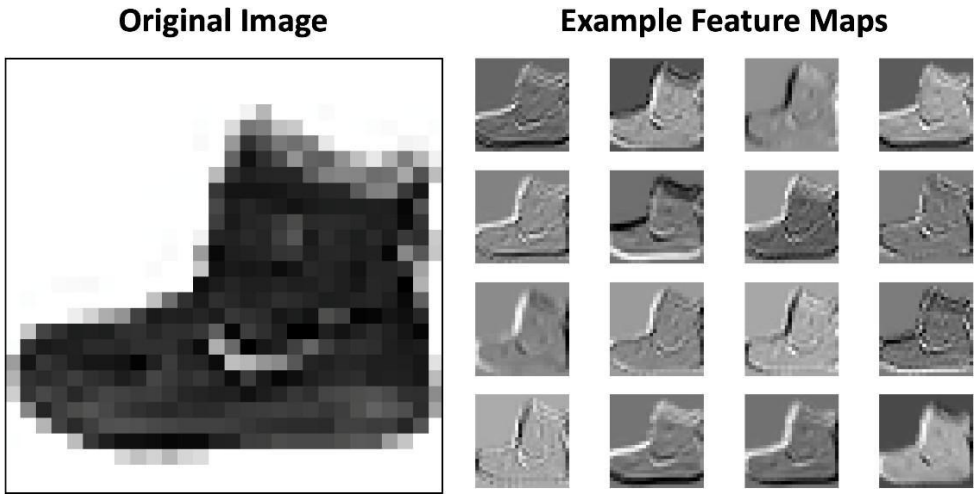
Εικόνα 2.2: Περιγραφή του συνελικτικού επιπέδου σε ένα ΣΝΔ και τη διαδικασία της συνέλευσης.

Στο (β) φαίνεται η συνέλιξη με φίλτρο 3x3: κάθε στοιχείο του φίλτρου πολλαπλασιάζεται με το αντίστοιχο στοιχείο της εικόνας και τα γινόμενα αθροίζονται για να δημιουργηθεί μια τιμή. Αυτή η τιμή αντικαθιστά την τιμή του κεντρικού πίξελ στη θέση εφαρμογής του φίλτρου (κόκκινο κελί). Η πράξη της συνέλιξης εφαρμόζεται με σταθερό βήμα (stride) 1.

2.2.2 Χάρτες Χαρακτηριστικών (Feature Maps)

Ο χάρτης χαρακτηριστικών (feature map) είναι το αποτέλεσμα που προκύπτει μετά από το συνελικτικό επίπεδο και δείχνει κάποια συγκεκριμένα χαρακτηριστικά ή μοτίβα που έχει εντοπίσει το δίκτυο σε μία είσοδο (π.χ. μια εικόνα). Κατά τη διάρκεια της προώθησης «σκανάρετε» ένα φίλτρο πάνω από την εικόνα εισόδου, μετακινείται πάνω της εστιάζοντας σε κάθε μικρή περιοχή ξεχωριστά. Για κάθε θέση του φίλτρου υπολογίζεται το γινόμενο του φίλτρου στην τοπική περιοχή της εικόνας και με την εφαρμογή μιας συνάρτησης ενεργοποίησης, προκύπτουν οι τελικές τιμές των χαρτών χαρακτηριστικών.

Κάθε φίλτρο δημιουργεί έναν χάρτη που τονίζει ένα συγκεκριμένο στοιχείο της εικόνας, όπως γραμμές ή υφές. Τελικά, οι χάρτες χαρακτηριστικών από τα διάφορα φίλτρα μπορούν να στοιβάζονται, και ο συνδυασμός αυτός επιτρέπει στο ΣΝΔ να κατανοεί πολύπλοκα και πολυδιάστατα μοτίβα σε μια εικόνα, δημιουργώντας μία πιο ολοκληρωμένη αναπαράσταση των δεδομένων εισόδου. [15]



Εικόνα 2.3:Παραδείγματα χαρτών χαρακτηριστικών που δημιουργούνται από ένα προεκπαιδευμένο μοντέλο.

2.3 Υπερ-παράμετροι των Νευρωνικών Επιπέδων (Hyperparameters)

Οι υπερ-παράμετροι των συνελικτικών νευρωνικών δικτύων, όπως το padding, ο διασκελισμός (stride) και τα φίλτρα είναι σημαντικοί παράμετροι που καθορίζουν το μέγεθος και την ανάλυση των χαρτών χαρακτηριστικών. Παίζουν πολύ σημαντικό ρόλο καθώς ελέγχουν τον όγκο των πληροφοριών και επηρεάζουν σημαντικά την απόδοση του δικτύου.

Ο όρος **stride** αναφέρεται στο βήμα που κάνει το φίλτρο (filter ή kernel) κάθε φορά που μετακινείται πάνω στην εικόνα εισόδου. Αν, για παράδειγμα, ορίσουμε το stride ως 1, τότε το φίλτρο μετακινείται ένα pixel τη φορά. Ένα μεγαλύτερο βήμα οδηγεί σε χαμηλότερη ανάλυση εξόδου, ενώ ένα μικρότερο βήμα οδηγεί σε υψηλότερη ανάλυση εξόδου. [17]

Το **padding** αφορά την προσθήκη επιπλέον γραμμών και στηλών (με pixels) γύρω από την εικόνα εισόδου, πριν την εφαρμογή των φίλτρων. Αυτή η προσθήκη συνήθως γίνεται με μηδενικά (zero padding), δηλαδή η τιμή που δίνεται είναι το 0. Αυτό βοηθάει στη διατήρηση των χωρικών διαστάσεων της εικόνας καθώς περνά μέσα από τα συνελικτικά στρώματα. Επίσης, για να διασφαλίσει ότι οι χάρτες εξόδου έχουν τις ίδιες χωρικές διαστάσεις με την εικόνα εισόδου ή για τη ρύθμιση του μεγέθους των χαρτών εξόδου. [15]

2.4 Συναρτήσεις Ενεργοποίησης (Activation Functions)

Οι συναρτήσεις ενεργοποίησης στα ΣΝΔ αναφέρονται στις μη-γραμμικές συναρτήσεις που εφαρμόζονται σε κάθε νευρώνα του δικτύου ξεχωριστά. Ο σκοπός τους είναι να εισάγουν μη-γραμμικότητα, επιτρέποντας στο δίκτυο να μάθει περίπλοκα μοτίβα. Στον πραγματικό κόσμο πολλά φαινόμενα χαρακτηρίζονται από πολύπλοκες και μη γραμμικές συμπεριφορές που δεν μπορούν να παρασταθούν με ακρίβεια από μοντέλα που χρησιμοποιούν μόνο γραμμικές συναρτήσεις. Γι' αυτόν το λόγο ο ρόλος τους είναι κρίσιμος. [18] Μερικές από τις πιο γνωστές συναρτήσεις είναι οι εξής:

- **Σιγμοειδής συνάρτηση**

Η σιγμοειδής συνάρτηση (sigmoid) είναι μία μη γραμμική συνάρτηση ενεργοποίησης. Η καμπύλη του μοιάζει με το γράμμα «S» και αναπαράγει μια έξοδο μεταξύ του 0 και 1. Η μαθηματική της έκφραση ορίζεται ως:

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

- **Υπερβολική εφαπτομένη συνάρτηση**

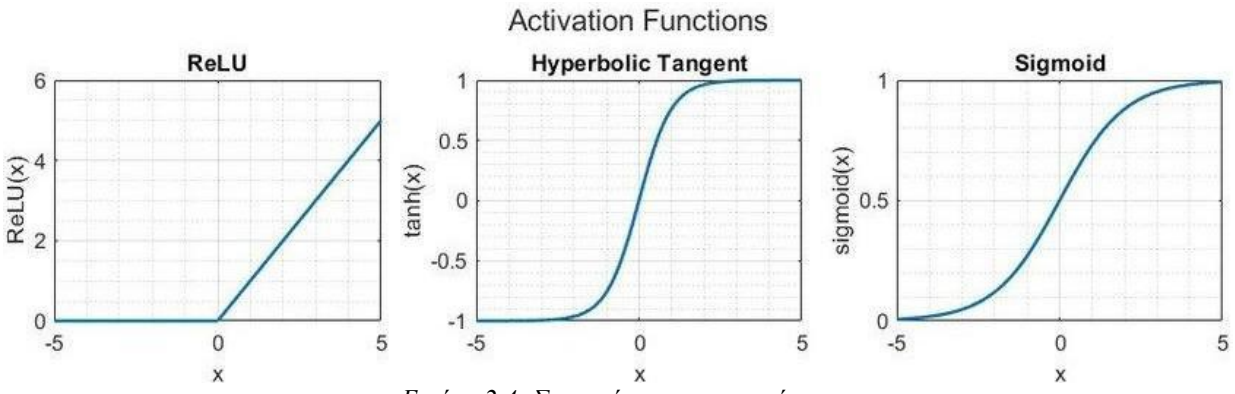
Η υπερβολική εφαπτομένη συνάρτηση (tanh) είναι επίσης μία μη γραμμική συνάρτηση χρήσιμη για την εισαγωγή μη γραμμικότητας στο μοντέλο. Είναι παρόμοια με την σιγμοειδή, αλλά χαρτογραφεί οποιονδήποτε πραγματικό αριθμό μεταξύ του -1 και 1. Η μαθηματική της συνάρτηση ορίζεται ως:

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

- **Rectified Linear Unit συνάρτηση**

Η συνάρτηση Rectified Linear Unit (ReLU) είναι η πιο δημοφιλής συνάρτηση ενεργοποίησης που θα δώσει έξοδο μόνο αν η είσοδος είναι θετική, διαφορετικά θα δώσει μηδενική έξοδο. Η μαθηματική της έκφραση ορίζεται ως:

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



Εικόνα 2.4: Συναρτήσεις ενεργοποιήσης.

Οι συναρτήσεις ενεργοποίησης sigmoid και tanh παρουσιάζουν ένα βασικό μειονέκτημα, το φαινόμενο της εξαφάνισης κλίσης (vanishing gradient problem). Αυτό το φαινόμενο υποδεικνύει ότι οι κλίσεις που χρησιμοποιούνται κατά την εκπαίδευση ενός νευρωνικού δικτύου τείνουν να γίνονται εξαιρετικά μικρές στα βαθιά επίπεδα, προκαλώντας επιβράδυνση και επηρεάζοντας την απόδοση του δικτύου. [19] Αυτό το πρόβλημα λύνει η συνάρτηση ReLU, η οποία λόγω της απλότητάς της διευκολύνει την ταχύτερη εκπαίδευση. Μία παραλλαγή αυτής είναι η leaky ReLU, μια ακόμα πιο προηγμένη συνάρτηση όπου δεν μηδενίζει εντελώς τις αρνητικές τιμές αλλά τις μειώνει βοηθώντας τους νευρώνες να ενεργοποιούνται ακόμα και για αρνητικές εισόδους. Σε αντίθεση με την «απλή» ReLU, αυτή η προσέγγιση εξαλείφει το πρόβλημα γνωστό ως «Dying ReLU», όπου πολλοί νευρώνες παραμένουν ανενεργοί λόγω μηδενικών εξόδων. [15]

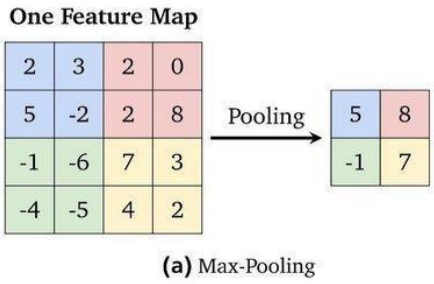
Συναρτήσεις Ενεργοποίησης	**Μαθηματικός τύπος**	**Περιγραφή**
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	Μη-γραμμική συνάρτηση μεταξύ 0 και 1.
Tanh	$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	Μη-γραμμική συνάρτηση μεταξύ -1 και 1.
ReLU	$f(x) = \max(0, x)$	Δίνει θετική έξοδο, αλλιώς 0.
Leaky ReLU	$f(x) = \{((x, if\ x > 0\ @\ ax\ if\ x < 0))\}$	Δίνει θετική έξοδο και διατηρεί μικρές αρνητικές τιμές.

Πίνακας 2.1: Συναρτήσεις ενεργοποίησης.

2.5 Συγκεντρωτικό Επίπεδο (Pooling Layer)

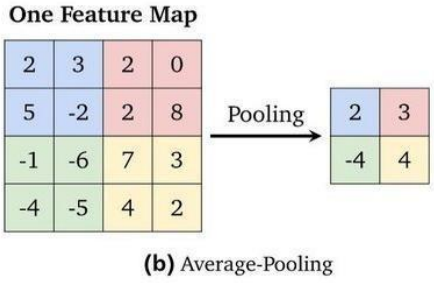
Το συγκεντρωτικό επίπεδο (pooling layer) συνήθως εφαρμόζεται μετά το συνελικτικό επίπεδο. Ο βασικός του ρόλος είναι η μείωση των χωρικών διαστάσεων των χαρτών χαρακτηριστικών περιορίζοντας έτσι τον όγκο των δεδομένων και τον αριθμό των παραμέτρων. Αυτό το βήμα είναι σημαντικό διότι συμβάλλει στην αποφυγή της υπερπροσαρμογής (overfitting) αλλά και στη μείωση του χρόνου εκπαίδευσης. Το συγκεντρωτικό επίπεδο λειτουργεί διαχωρίζοντας κάθε χάρτη χαρακτηριστικών σε μικρότερες περιοχές σταθερού μεγέθους και εφαρμόζοντας μία από τις δύο βασικές λειτουργίες συγκέντρωσης, ανάλογα με τις ανάγκες κάθε δικτύου: τη μέγιστη συγκέντρωση (max pooling) ή την μέση συγκέντρωση (average pooling).

Στη μέγιστη συγκέντρωση (max pooling) το σύστημα μειώνει τις χωρικές διαστάσεις των χαρακτηριστικών επιλέγοντας τη μέγιστη τιμή μέσα σε κάθε μικρό παράθυρο ή περιοχή. Συγκεκριμένα, για κάθε τέτοια περιοχή, επιλέγεται η μεγαλύτερη τιμή και αντικαθιστά τα αρχικά δεδομένα μαζί του. Αυτή η λειτουργία βοηθά στη διατήρηση των πιο έντονων χαρακτηριστικών σε μια εικόνα, όπως οι άκρες και υφές, επιτρέποντας στο δίκτυο να εστιάζει στις πιο σημαντικές πληροφορίες. [20]



Εικόνα 2.5: Εφαρμογή μέγιστης συγκέντρωσης σε έναν χάρτη χαρακτηριστικών 4x4 με φίλτρο 2x2, που παράγει έξοδο 2x2.

Στη μέση συγκέντρωση (average pooling) το σύστημα μειώνει τις χωρικές διαστάσεις των χαρακτηριστικών υπολογίζοντας τον μέσο όρο μέσα σε κάθε μικρό παράθυρο ή περιοχή. Συγκεκριμένα, για κάθε τέτοια περιοχή, επιλέγεται η μέση τιμή και αντικαθιστά τα αρχικά δεδομένα μαζί του. Ωστόσο, αυτή η μέθοδος μπορεί να μην είναι τόσο αποτελεσματική στην ανίχνευση έντονων χαρακτηριστικών, καθώς εξομαλύνει τις τιμές και μειώνει την ανάλυση των λεπτομερειών. [20]

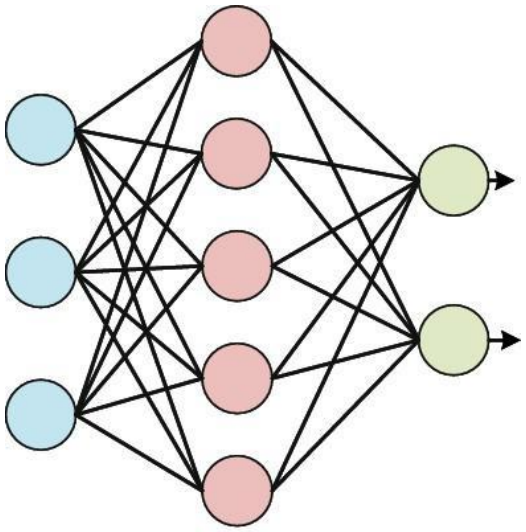


Εικόνα 2.6: Εφαρμογή μέσης συγκέντρωσης σε έναν χάρτη χαρακτηριστικών 4x4 με φίλτρο 2x2, που παράγει έξοδο 2x2.

2.6 Πλήρως Συνδεδεμένο Επίπεδο (Fully Connected Layer)

Το πλήρως συνδεδεμένο επίπεδο βρίσκεται στο τέλος του δικτύου και είναι υπεύθυνο για την ολοκλήρωση των εργασιών ταξινόμησης και παλινδρόμησης. Σε αυτό το στρώμα, οι χάρτες χαρακτηριστικών που έχουν παραχθεί από τα συνελικτικά και τα συγκεντρωτικά επίπεδα, μετατρέπονται σε μονοδιάστατα διανύσματα (flattening), που μπορούν να συνδεθούν με τους νευρώνες εξόδου. Κάθε νευρώνας στο πλήρως συνδεδεμένο επίπεδο είναι συνδεδεμένος με όλους τους νευρώνες του προηγούμενου επιπέδου, αυξάνοντας έτσι τον αριθμό των παραμέτρων.

Για την εισαγωγή μη γραμμικών χαρακτηριστικών στο δίκτυο χρησιμοποιούνται συναρτήσεις ενεργοποίησης (activation functions), οι οποίες αυξάνουν την ικανότητα του δικτύου να αναπαριστά σύνθετα μοτίβα. Τα πλήρως συνδεδεμένα στρώματα τείνουν να υπερπροσαρμόζονται στα δεδομένα εκπαίδευσης. Για την αντιμετώπιση αυτού του φαινομένου, συχνά συνδυάζονται με τεχνικές κανονικοποίησης που αποσκοπούν στη βελτίωση της ικανότητας γενίκευσης του δικτύου. [18]



Εικόνα 2.7: Πλήρως συνδεδεμένο επίπεδο.

2.7 Εκπαίδευση

2.7.1 Συνάρτηση Κόστους

Η συνάρτηση κόστους (loss function) είναι ένα μαθηματικό εργαλείο που αξιολογεί πόσο καλά αποδίδει το δίκτυο, χρησιμοποιώντας δεδομένα εκπαίδευσης. Χρησιμοποιείται για τη μέτρηση της διαφοράς μεταξύ των προβλέψεων του μοντέλου και των πραγματικών τιμών, αποτελώντας τον πυρήνα των αλγορίθμων βελτιστοποίησης. Με την ελαχιστοποίηση της συνάρτησης κόστους, οι παράμετροι του μοντέλου προσαρμόζονται ώστε οι προβλέψεις του να πλησιάζουν όσο το δυνατόν περισσότερο τις πραγματικές τιμές. [18]

Οι πιο συνηθισμένες συναρτήσεις κόστους είναι αυτή του μέσου τετραγωνικού σφάλματος (Mean squared error, MSE), που εφαρμόζεται σε προβλήματα παλινδρόμησης, και η συνάρτηση εντροπίας (Cross-entropy loss, CE) η οποία χρησιμοποιείται κυρίως σε προβλήματα ταξινόμησης.

Συνάρτηση μέσου τετραγωνικού σφάλματος:

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

Συνάρτηση εντροπίας:

$$CE = -\frac{1}{n}\sum_{i=1}^{n}[y_i\, log(\hat{y}_i) + (1 - y_i)\, log(1 - \hat{y}_i)]$$

Όπου:

- n: πλήθος δεδομένων
- y_i: η πραγματική τιμή του i-οστού δείγματος
- y_i': η προβλεπόμενη τιμή για το i-οστό δείγμα από το μοντέλο
- i: δείκτης των δειγμάτων

2.7.2 Οπισθοδιάδοση

Η οπισθοδιάδοση (backpropagation) είναι μια βασική διαδικασία κατά την εκπαίδευση ενός συνελικτικού νευρωνικού δικτύου, η οποία στοχεύει στη βελτιστοποίηση των παραμέτρων του, μέσω της ελαχιστοποίησης της συνάρτησης κόστους. Ο πυρήνας της διαδικασίας είναι ο υπολογισμός της κλίσης της συνάρτησης κόστους σε σχέση με τις παραμέτρους του δικτύου, όπως τα βάρη και οι πολώσεις (biases). Με τη χρήση αλγορίθμων βελτιστοποίησης, τα βάρη ενημερώνονται έτσι ώστε η τιμή της συνάρτησης κόστους να μειώνεται σταδιακά.

Η διαδικασία ξεκινά με την προώθηση των δεδομένων μέσω του δικτύου (forward propagation), όπου υπολογίζεται η έξοδος του δικτύου και η τιμή της συνάρτησης κόστους. Στη συνέχεια, μέσω της ανάστροφης διάδοσης, το σφάλμα διαδίδεται από το επίπεδο εξόδου προς τα πίσω στα προηγούμενα επίπεδα, υπολογίζοντας το πώς επηρεάζει κάθε παράμετρο η συνάρτηση κόστους. Αυτό επιτυγχάνεται με τη χρήση του κανόνα της αλυσίδας, που επιτρέπει τον υπολογισμό των παραγώγων σε κάθε επίπεδο. Οι κλίσεις αυτές αξιοποιούνται από τον αλγόριθμο βελτιστοποίησης για την ενημέρωση των βαρών, επαναλαμβάνοντας τη διαδικασία έως ότου το δίκτυο επιτύχει ικανοποιητική απόδοση. [15] [18] [21]

2.7.3 Αλγόριθμοι Βελτιστοποίησης

Οι αλγόριθμοι βελτιστοποίησης (optimization algorithms) είναι ένα εργαλείο των ΣΝΔ οι οποίοι χρησιμοποιούνται για την ενημέρωση των βαρών και των πολώσεων (biases) κατά την διάρκεια της εκπαίδευσης για την ελαχιστοποίηση της συνάρτησης κόστους. [17] Η κύρια ιδέα των αλγορίθμων αυτών είναι να υπολογίζουν την κλίση (gradient) της συνάρτησης κόστους σε σχέση με τις παραμέτρους και να προσαρμόζει τις τιμές αυτές με έναν τρόπο που μειώνει το σφάλμα. [18] Παρακάτω θα δούμε μερικούς από αυτούς τους αλγόριθμους αναλυτικά.

- **Αλγόριθμος μείωσης κλίσης (gradient descent)**

Η μέθοδος αυτή αποτελεί τον πιο θεμελιώδη αλγόριθμο βελτιστοποίησης, ο οποίος προσαρμόζει τις παραμέτρους με βάση την κατεύθυνση της κλίσης της συνάρτησης κόστους, μειώνοντας σταδιακά την συνάρτηση κόστους. (8) Οι πιο συνηθισμένοι αλγόριθμοι μείωσης κλίσης είναι ο αλγόριθμος στοχαστικής μείωσης κλίσης (stochastic gradient descent), μείωση κλίσης με πλήρη παρτίδα (batch gradient descent) και μείωση κλίσης με μίνι-παρτίδα (mini-batch gradient descent). Πολλά κλασσικά μοντέλα ΣΝΔ όπως είναι τα AlexNet, VGG, ResNet και FaceNet έχουν χρησιμοποιήσει αυτούς τους αλγόριθμους για να εκπαιδεύσουν τα δίκτυά τους. [22]

- **Ορμή (Momentum)**

  Η ορμή είναι μία μέθοδος όπου προσομοιώνει την φυσική έννοια της ορμής, χρησιμοποιώντας τον εκθετικά σταθμισμένο μέσο όρο της κλίσης για την ενημέρωση των βαρών. Βασικό πλεονέκτημά είναι η ικανότητά της να αποτρέπει ταλαντώσεις κατά τη διάρκεια της εκπαίδευσης, ειδικά όταν η κλίση σε μία διάσταση είναι πολύ μεγαλύτερη από την κλίση σε άλλη διάσταση. Έτσι, επιτυγχάνεται ταχύτερη σύγκλιση. [22] Η προσέγγιση αυτή εισάγει έναν συντελεστή ορμής που παρακολουθεί ιστορικά τις πληροφορίες των κλίσεων, λαμβάνοντας υπόψη τις προηγούμενες κατευθύνσεις ενημέρωσης των παραμέτρων. Αυτό επιτρέπει στο μοντέλο να επιταχύνει τη σύγκλιση και να μειώσει τις ταλαντώσεις, κάνοντάς το πιο σταθερό και αποτελεσματικό κατά την εκπαίδευση. [18]

- **Adagrad**

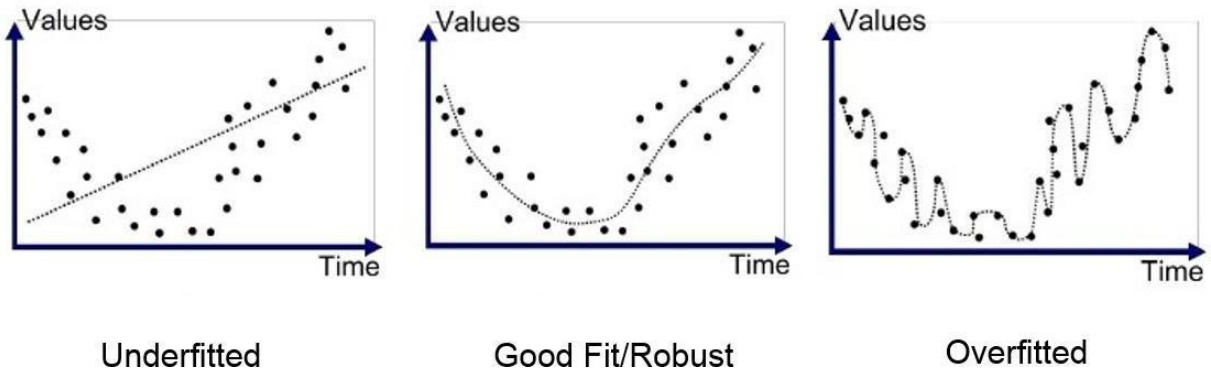
  Ο αλγόριθμος Adagrad είναι μια μέθοδος βελτιστοποίησης ο οποίος έχει τη δυνατότητα να προσαρμόζει δυναμικά το ρυθμό εκπαίδευσης για κάθε παράμετρο. Ένα από τα κύρια πλεονεκτήματα της μεθόδου αυτής είναι ότι δεν απαιτεί χειροκίνητη ρύθμιση του ρυθμού εκπαίδευσης. Ο αλγόριθμος προσαρμόζει τον ρυθμό εκπαίδευσης για κάθε βάρος με βάση το μέγεθος των κλίσεων που έχουν παρατηρηθεί έως εκείνο το σημείο. Έτσι, επιτυγχάνεται ταχύτερη σύγκλιση σε "επίπεδες" περιοχές της συνάρτησης κόστους και πιο αργή σύγκλιση σε "απότομες" περιοχές. [22]

- **Adam**

  Ο Adam (Adaptive Moment Estimation) είναι ένας αλγόριθμος βελτιστοποίησης που συνδυάζει ιδέες από την ορμή (momentum) και τον αλγόριθμο Adagrad δημιουργώντας έναν προσαρμοστικό αλγόριθμο ρυθμού εκπαίδευσης. Αυτός ο συνδυασμός του επιτρέπει να λειτουργεί αποδοτικά σε ένα ευρύ φάσμα νευρωνικών δικτύων, καθιστώντας τον εξαιρετικά αποτελεσματικό στην πράξη. Λαμβάνοντας υπόψη τις πρώτες και δεύτερες στατιστικές στιγμές των κλίσεων, ο Adam επιτυγχάνει σταθερότητα και ταχύτητα στη σύγκλιση, γεγονός που εξηγεί την ευρεία χρήση του στη βαθιά μάθηση. [18]

2.8 Πρόβλημα Υπερπροσαρμογής

Κατά τη δημιουργία μοντέλων ΣΝΔ, η υπερπροσαρμογή (overfitting) αποτελεί ένα από τα βασικότερα εμπόδια. Η υπερπροσαρμογή περιγράφει μια κατάσταση κατά την οποία ένα μοντέλο αποδίδει εξαιρετικά καλά στα δεδομένα εκπαίδευσης αλλά αποτυγχάνει να γενικεύσει σε νέα δεδομένα, όπως τα δεδομένα ελέγχου (test data). Αντίθετα, όταν το μοντέλο δεν μαθαίνει αρκετές πληροφορίες από τα δεδομένα εκπαίδευσης θεωρείται ότι παρουσιάζει υποπροσαρμογή (underfitting). Το μοντέλο θεωρείται κατάλληλο όταν αποδίδει ικανοποιητικά τόσο στα δεδομένα εκπαίδευσης όσο και στα δεδομένα ελέγχου, επιτυγχάνοντας έτσι μια καλή ισορροπία μεταξύ εκπαίδευσης και γενίκευσης. [15]



Εικόνα 2.8: Γραφικές παραστάσεις που παρουσιάζουν το πρόβλημα της υπερπροσαρμογής.

Στην πρώτη γραφική παράσταση το μοντέλο δεν έχει εκπαιδευτεί αρκετά, στην δεύτερη το μοντέλο θεωρείται κατάλληλο και στην τελευταία το μοντέλο έχει εκπαιδευτεί πάρα πολύ καλά και αποτυγχάνει να γενικεύσει.

2.9 Τεχνικές Κανονικοποίησης

Οι τεχνικές κανονικοποίησης χρησιμοποιούνται για την αντιμετώπιση του overfitting στα νευρωνικά δίκτυα, ενός φαινομένου κατά το οποίο το μοντέλο επιτυγχάνει υψηλή ακρίβεια στα δεδομένα εκπαίδευσης, αλλά αποτυγχάνει να γενικεύσει σε νέα, αόρατα δεδομένα. Οι τεχνικές αυτές λειτουργούν προσθέτοντας έναν όρο ποινής στη συνάρτηση κόστους. Αυτός ο όρος ενθαρρύνει το μοντέλο να έχει απλούστερα βάρη ή να μειώσει το μέγεθος των βαρών. [17]

- **Περιορισμός ενεργοποίησης (Dropout)**

Στην συγκεκριμένη τεχνική γίνεται τυχαία «απενεργοποίηση» ορισμένων νευρώνων κατά τη διάρκεια της εκπαίδευσης, γεγονός που βοηθά στην αποτροπή του μοντέλου να εξαρτάται υπερβολικά από συγκεκριμένα χαρακτηριστικά και μπορεί να αυξήσει την απόδοση γενίκευσης. Ο περιορισμός ενεργοποίησης εφαρμόζεται κάνοντας την έξοδο κάθε νευρώνα μηδενική με μια συγκεκριμένη πιθανότητα p. [23]

- **Πρόωρη διακοπή εκπαίδευσης (Early stopping)**

Σε αυτή την τεχνική διακόπτεται η διαδικασία εκπαίδευσης πριν το μοντέλο αρχίσει να υπερπροσαρμόζεται. Η ιδέα είναι να παρακολουθείται η απόδοση του μοντέλου σε ένα σύνολο επικύρωσης (validation set) κατά τη διάρκεια της εκπαίδευσης και να σταματάει όταν η απόδοση αυτή αρχίζει να μειώνεται. Αυτό υποδεικνύει ότι το μοντέλο ξεκινά να προσαρμόζεται υπερβολικά στα δεδομένα εκπαίδευσης. Το σύνολο επικύρωσης είναι ένα υποσύνολο των δεδομένων που δεν χρησιμοποιείται για εκπαίδευση αλλά για την αξιολόγηση της γενίκευσης του μοντέλου, ώστε να ανιχνευθεί η πιθανή υπερπροσαρμογή. [24]

- **Ομαλοποίηση L1 και L2 (Regularization)**

Οι τεχνικές αυτές προσθέτουν ποινές στη συνάρτηση κόστους, περιορίζοντας τα βάρη του μοντέλου ώστε να αποτρέπονται υπερβολικά μεγάλοι συντελεστές, οι οποίοι θα μπορούσαν να οδηγήσουν σε υπερπροσαρμογή.

Η L1 ομαλοποίηση ελαχιστοποιεί το άθροισμα των απόλυτων τιμών των βαρών, γεγονός που προάγει τη σπανιότητα, δηλαδή την επιλογή μόνο των πιο σημαντικών χαρακτηριστικών, αγνοώντας τα υπόλοιπα. Αυτό την καθιστά ιδανική για περιπτώσεις όπου μόνο λίγα χαρακτηριστικά είναι ουσιαστικά για την εκμάθηση του μοντέλου. Από την άλλη, η L2 ομαλοποίηση ελαχιστοποιεί το άθροισμα των τετραγώνων των βαρών (ridge regression), συρρικνώνοντας ομαλά όλα τα βάρη χωρίς να τα μηδενίζει, γεγονός που είναι ιδιαίτερα χρήσιμο όταν τα χαρακτηριστικά είναι συσχετισμένα. [25]

3. Μεθοδολογία

3.1 Επισκόπηση

Σε αυτή την ενότητα περιγράφεται λεπτομερώς η διαδικασία ανάπτυξης ενός συστήματος ανίχνευσης βίντεο deepfake χρησιμοποιώντας μια προσέγγιση που βασίζεται στη βαθιά μάθηση. Ο πρωταρχικός στόχος ήταν να δημιουργηθεί ένας ισχυρός ταξινομητής, ο οποίος θα είναι ικανός να διακρίνει πραγματικά και ψεύτικα καρέ βίντεο με μεγάλη ακρίβεια.

Η μεθοδολογία περιλαμβάνει τη συλλογή και προεπεξεργασία δεδομένων, την επιλογή κατάλληλου μοντέλου βαθιάς μάθησης και την εκπαίδευση του μοντέλου μέσω καθορισμένων μετρικών. Η προσέγγιση βασίστηκε στη χρήση βιβλιοθηκών όπως η **Pytorch** και η **Timm** για την υλοποίηση και εκπαίδευση του μοντέλου, ενώ τα δεδομένα συλλέχθηκαν από καθιερωμένα σύνολα δεδομένων ανίχνευσης deepfake.

Η δημιουργία ενός αξιόπιστου ταξινομητή deepfake απαιτεί την αντιμετώπιση προκλήσεων όπως η ποικιλομορφία των δεδομένων, η υψηλή υπολογιστική πολυπλοκότητα και η ανάγκη για ανίχνευση μικρών δεδομένων στα καρέ βίντεο. Το τελικό μοντέλο στοχεύει να παρέχει μια λύση που μπορεί να ενσωματωθεί σε πραγματικές εφαρμογές για την ανίχνευση deepfake βίντεο.

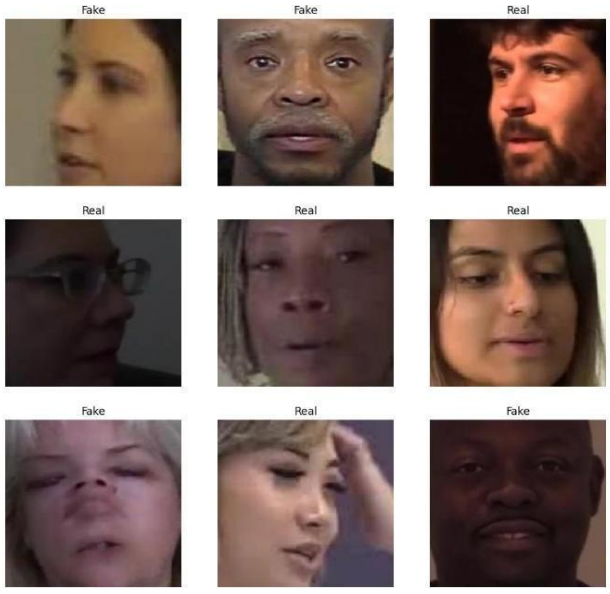
3.2 Σύνολο Δεδομένων και Επεξεργασία

3.2.1 Περιγραφή Συνόλου Δεδομένων

Στην παρούσα μελέτη έχουμε χρησιμοποιήσει τρία διαφορετικά σύνολα δεδομένων τα οποία περιέχουν επισημασμένα πραγματικά και ψεύτικα καρέ βίντεο. Αναλυτικότερα χρησιμοποιήθηκαν βίντεο στα οποία έχει γίνει ανίχνευση προσώπων και στη συνέχεια πραγματοποιήθηκε δειγματοληψία του μεσαίου frame, προκειμένου να διευκολυνθεί η ταξινόμηση σε επίπεδο εικόνας. Το σύνολο δεδομένων χωρίστηκε σε σύνολο εκπαίδευσης και δοκιμής με αναλογία 80:20 για να εξασφαλιστούν επαρκή δεδομένα για την εκπαίδευση διατηρώντας παράλληλα ένα αξιόπιστο σύνολο ελέγχου για την αξιολόγηση της απόδοσης. Παρακάτω θα δούμε συγκεκριμένα τα σύνολα δεδομένων που έχουν χρησιμοποιηθεί:

- **DFDC**

Το DFDC dataset (DeepFake Detection Challenge) είναι ένα από τα μεγαλύτερα δημόσια διαθέσιμα σύνολα δεδομένων deepfake, το οποίο χρησιμοποιεί μια ποικιλία μεθόδων δημιουργίας ψεύτικων βίντεο, συμπεριλαμβανομένων τόσο προηγμένων τεχνικών βασισμένων σε GANs όσο και απλούστερων προσεγγίσεων, με στόχο την εκπαίδευση αποτελεσματικών μοντέλων ανίχνευσης. [26] Η έκδοση του DFDC που χρησιμοποιήθηκε περιέχει 1500 ψεύτικες εικόνες και 1500 πραγματικές εικόνες, εξάγοντας ένα καρέ ανά βίντεο και απομονώνοντας τα πρόσωπα κάθε ατόμου από τα βίντεο. Με αυτό τον τρόπο διασφαλίστηκε ένα ισορροπημένο σύνολο δεδομένων για εκπαίδευση και αξιολόγηση, το οποίο θα έχει σαν αποτέλεσμα μια πιο αντιπροσωπευτική και δίκαιη μέτρηση της ακρίβειας.



Εικόνα 3.1: Οπτικοποίηση δείγματος εικόνων από το σύνολο δεδομένων DFDC.

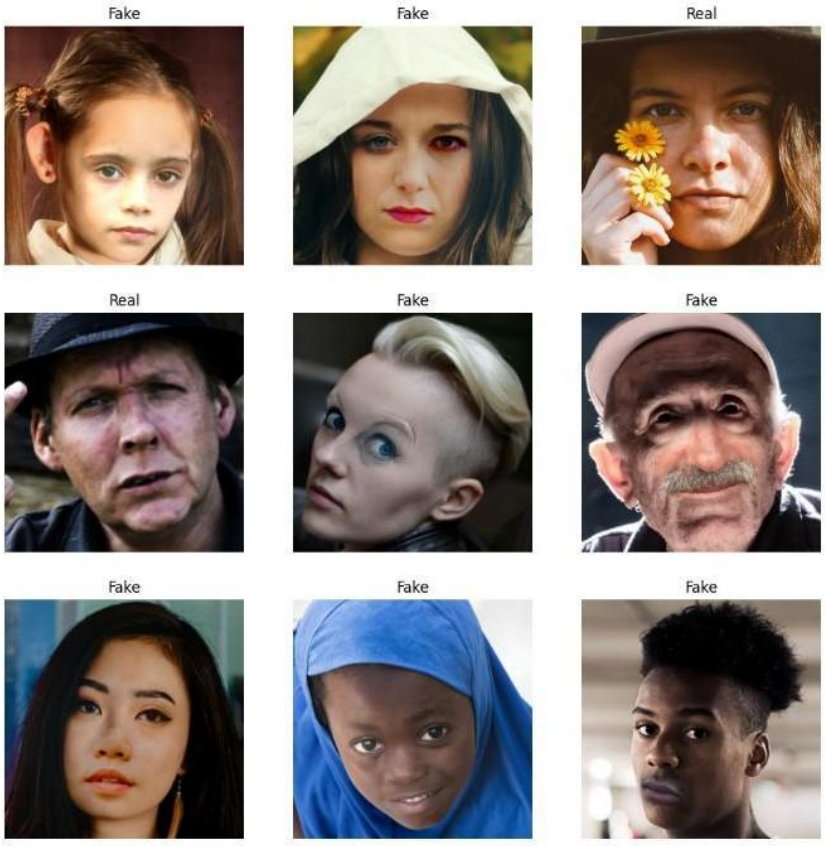
- **DFMNIST+**

Το **Deepfake MNIST+** αποτελεί ένα προηγμένο σύνολο δεδομένων που επικεντρώνεται στις κινούμενες εκφράσεις προσώπου αντί για την ανταλλαγή ταυτότητας, καλύπτοντας ένα σημαντικό κενό στην έρευνα για την ανίχνευση deepfake. Δημιουργήθηκε με τη χρήση σύγχρονων πλαισίων animation εικόνας, προσομοιώνοντας 10 συγκεκριμένες κινήσεις, όπως το ανοιγοκλείσιμο των ματιών, την έκπληξη και την αμηχανία, και αποδεικνύεται ικανό να ξεγελάσει κορυφαία συστήματα ανίχνευσης ζωντάνιας. Τα πειραματικά δεδομένα δείχνουν ότι τα υπάρχοντα μοντέλα που έχουν εκπαιδευτεί σε σύνολα δεδομένων ανταλλαγής ταυτότητας παρουσιάζουν χαμηλή απόδοση στην ανίχνευση αυτών των animation, ενώ η εκπαίδευση σε συνδυασμό με το DFMNIST+ βελτιώνει σημαντικά την απόδοση. [27] Επιπλέον, για να αποφευχθεί η μεροληψία στην εκπαίδευση, διασφαλίστηκε ότι κάθε δείγμα αντιπροσωπεύει ένα μοναδικό άτομο, αναγκάζοντας το μοντέλο να διακρίνει πραγματικά μεταξύ αυθεντικών και ψεύτικων βίντεο.

Εικόνα 3.2: Οπτικοποίηση δείγματος εικόνων από το σύνολο δεδομένων DFMNIST+.

- **EXP**

Το **EXP dataset** αποτελεί μια μοναδική συλλογή επεξεργασμένων εικόνων προσώπων, που δημιουργήθηκαν από ειδικούς του Computational Intelligence and Photography Lab του Πανεπιστημίου Yonsei, χρησιμοποιώντας προηγμένα εργαλεία όπως το Photoshop. Περιλαμβάνει περίπου 1.000 αυθεντικές και 1.000 επεξεργασμένες εικόνες, όπου τα χαρακτηριστικά των προσώπων, όπως τα μάτια, η μύτη και το στόμα, έχουν αντικατασταθεί ή τροποποιηθεί για να συνθέσουν νέες εικόνες. Το EXP προσφέρει δεδομένα υψηλής ποιότητας που μπορούν να εκπαιδεύσουν ταξινομητές ικανούς να ανιχνεύσουν ψεύτικες εικόνες ανεξάρτητα από τη μέθοδο δημιουργίας τους. [28]



Εικόνα 3.3: Οπτικοποίηση δείγματος εικόνων από το σύνολο δεδομένων EXP.

3.3 Προεπεξεργασία Δεδομένων

Για την επεξεργασία των δεδομένων χρησιμοποιήθηκε η συνάρτηση **transform.Compose**, η οποία ανήκει στη βιβλιοθήκη **torchvision.transforms**. Η συνάρτηση αυτή επιτρέπει τον συνδυασμό πολλών μετασχηματισμών στις εικόνες. Με άλλα λόγια, συνδυάζει μια λίστα από μετασχηματισμούς που εφαρμόζονται διαδοχικά σε κάθε εικόνα που περνά από το σύνολο δεδομένων. Στη συγκεκριμένη περίπτωση εφαρμόστηκαν οι εξής μετασχηματισμοί:

1. **Resize**: Τα πλαίσια εισόδου άλλαξαν μέγεθος σε 450x450 pixel για να τυποποιηθούν οι διαστάσεις σε όλα τα δείγματα.

2. **ToTensor**: Η εικόνα μετατρέπεται σε μορφή Tensor, ώστε να μπορεί να επεξεργαστεί από το νευρωνικό δίκτυο.

```
from torchvision import transforms


transform = transforms.Compose([
    transforms.Resize((450, 450)),  # input size
    transforms.ToTensor(),
])
```

Πίνακας 3.1: Ορισμός μετασχηματισμών για την προεπεξεργασία των εικόνων.

Επιπλέον, οι εικόνες κανονικοποιήθηκαν για να εξασφαλίσουν ένα συνεπές εύρος εντάσεων των pixels. Κατά τη διάρκεια της εκπαίδευσης χρησιμοποιήθηκε μέγεθος παρτίδας 32 για τη βελτιστοποίηση της χρήσης μνήμης GPU. Με τις κατάλληλες διαμορφώσεις μπορούμε να διασφαλίσουμε ότι οι εικόνες στο σύνολο δεδομένων είναι έτοιμες και σε σωστή μορφή για να τροφοδοτηθούν στο μοντέλο. Επίσης, εκχωρούμε δυαδικές ετικέτες σε πραγματικές (0) και ψεύτικες (1).

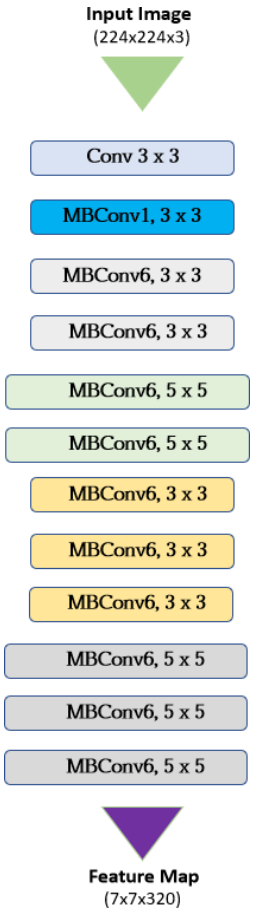
```
# --------------- DATASET BUILDER ---------------
class Dataset_Builder(Dataset):
    def __init__(self, fake_dir, real_dir, transform=None):
        self.transform = transform
        self.fake_images = [os.path.join(fake_dir, img) for img in
os.listdir(fake_dir)]
        self.real_images = [os.path.join(real_dir, img) for img in
os.listdir(real_dir)]
        self.all_images = self.fake_images + self.real_images
        self.labels = [1] * len(self.fake_images) + [0] *
len(self.real_images)  # 1 for fake, 0 for real


    def __len__(self):
        return len(self.all_images)


    def __getitem__(self, idx):
        image_path = self.all_images[idx]
        image = Image.open(image_path)
        if self.transform:
            image = self.transform(image)
        label = self.labels[idx]


        name_id = image_path.split('/')[-1].replace('.jpg', '')
        return image, label, name_id
```

Πίνακας 3.2: Δημιουργία συνόλου δεδομένων με κανονικοποίηση και ανάθεση δυαδικών ετικετών.

3.4 Αρχιτεκτονική Μοντέλου

3.4.1 Επιλογή Μοντέλου

Το EfficientNet-B0, ένα συνελικτικό νευρωνικό δίκτυο σχεδιασμένο για βέλτιστη απόδοση, επιλέχθηκε ως η βασική αρχιτεκτονική του μοντέλου. Χάρη στη δυνατότητα κλιμάκωσης που προσφέρει, αλλά και στα προεκπαιδευμένα βάρη του πάνω στο ImageNet, αποτέλεσε ένα ισχυρό σημείο εκκίνησης για μεταφορά μάθησης (transfer learning). Η αποτελεσματικότητα του EfficientNet έγκειται στον έξυπνο συνδυασμό πλάτους, βάθους και ανάλυσης, επιτρέποντας την επίτευξη υψηλής ακρίβειας με χαμηλές υπολογιστικές απαιτήσεις. Αυτή η ιδιότητα το καθιστά ιδανικό για εφαρμογές που απαιτούν ταχύτητα, ακρίβεια και αποδοτικότητα, όπως είναι η ανίχνευση deepfake. [29]



Εικόνα 3.5: Διαγραμματική απεικόνιση της αρχιτεκτονικής του EfficientNet.

3.4.2 Τροποποιήσεις Μοντέλου

Η κλάση **Network** ορίζει ένα νευρωνικό δίκτυο που χρησιμοποιεί το προεκπαιδευμένο μοντέλο **EfficientNet-B0** ως βασικό μοντέλο για την εξαγωγή χαρακτηριστικών. Η κλάση Network κληρονομεί από την κλάση **nn.Module** της PyTorch., η οποία αποτελεί τη βασική κλάση για την κατασκευή νευρωνικών δικτύων.

Στη στρατηγική που έχει επιλεγεί, αρχικά όλα τα στρώματα του EfficientNet-B0 "παγώνουν" (freezing) μέσω της παραμέτρου **requires_grad=False**, ώστε να διατηρηθούν οι αναπαραστάσεις που έχουν μάθει από την εκπαίδευση τους στο ImageNet. Αυτή η διαδικασία είναι κρίσιμη, καθώς τα πρώτα στρώματα του δικτύου περιέχουν γενικές αναπαραστάσεις και δεν χρειάζονται προσαρμογή. Επιπλέον, με το πάγωμα των στρωμάτων μειώνεται ο υπολογιστικός φόρτος κατά τη διάρκεια της εκπαίδευσης.

Για την επίτευξη της εξειδίκευσης, τα τελευταία πέντε στρώματα του EfficientNet-B0 ξεπαγώνουν (unfreezing). Αυτή η διαδικασία υλοποιείται μέσω της μεθόδου **unfreeze_last_layers(),** η οποία ενεργοποιεί την εκπαίδευση των τελευταίων στρωμάτων. Τα συγκεκριμένα στρώματα προσαρμόζονται στο τρέχον σύνολο δεδομένων, επιτρέποντας στο μοντέλο να αποδίδει καλύτερα στην εργασία ταξινόμησης εικόνων που έχει ανατεθεί.

Παράλληλα, το τελευταίο πλήρως συνδεδεμένο στρώμα του EfficientNet-B0 αντικαθίσταται με ένα γραμμικό στρώμα (linear layer), σχεδιασμένο για δυαδική ταξινόμηση (binary classification). Το νέο στρώμα λαμβάνει ως είσοδο τα χαρακτηριστικά που εξάγονται από το EfficientNet-B0 και επιστρέφει μία έξοδο, η οποία αντιπροσωπεύει την πιθανότητα να ανήκει σε αυτήν κατηγορία. Η διαδικασία της μερικής προσαρμογής των τελικών στρωμάτων (fine-tuning) εξασφαλίζει την ισορροπία ανάμεσα στη διατήρηση της γενικής γνώσης που έχει αποκτηθεί από το EfficientNet-B0 και στην εξειδίκευση του μοντέλου για την τρέχουσα εργασία.

Η συνολική αρχιτεκτονική του μοντέλου και οι τροποποιήσεις που εφαρμόστηκαν εξασφαλίζουν ένα αποδοτικό και ευέλικτο σύστημα ταξινόμησης, ικανό να προσαρμοστεί στις απαιτήσεις της εργασίας χωρίς να θυσιάζει τη γενική του ικανότητα.

```
class Network(nn.Module):
    def __init__(self, base_model_name="efficientnet_b0"):
        super(Network, self).__init__()

        base_model = timm.create_model(base_model_name, pretrained=True)
        self.features = base_model

        # Freeze initials layers
        for param in self.features.parameters():
            param.requires_grad = False

        # Adjust output layer for binary classification
        num_features = base_model.classifier.in_features
        base_model.classifier = nn.Linear(num_features, 1)

    def unfreeze_last_layers(self, num_layers=5):
        layers = list(self.features.children())
        for layer in layers[-num_layers:]:
            for param in layer.parameters():
                param.requires_grad = True

    def forward(self, x):
        x = self.features(x)
        return x
```

Πίνακας 3.4: Υλοποίηση της αρχιτεκτονικής του μοντέλου.

3.5 Περιβάλλον Εκπαίδευσης

- **Hardware**

Για την υλοποίηση και την εκπαίδευση του μοντέλου χρησιμοποιήθηκε η διαδικτυακή πλατφόρμα Kaggle, η οποία παρέχει υπολογιστικούς πόρους υψηλής απόδοσης για την ανάπτυξη και εκπαίδευση συνελικτικών νευρωνικών δικτύων. Η επιλογή του Kaggle ως περιβάλλον εκπαίδευσης έγινε διότι, προσφέρει πρόσβαση σε εξειδικευμένο υλικό και λογισμικό υψηλών επιδόσεων, όπως η GPU T4 της NVIDIA. Η συγκεκριμένη GPU είναι σχεδιασμένη για υψηλής απόδοσης υπολογιστικές εργασίες όπως η εκπαίδευση νευρωνικών δικτύων και η επεξεργασία μεγάλων δεδομένων. Η GPU T4 διαθέτει 16GB μνήμης GDDR6, γεγονός που την καθιστά ιδανική για εργασίες βαθιάς μάθησης, καθώς παρέχει την απαραίτητη μνήμη για τη διαχείριση μεγάλων δεδομένων και την αποδοτική εκτέλεση πολλαπλών μαθηματικών πράξεων. [30]

Στο πλαίσιο αυτής της διπλωματικής εργασίας, το Kaggle χρησιμοποιήθηκε ως κύριο περιβάλλον ανάπτυξης και εκπαίδευσης του μοντέλου. Το περιβάλλον αυτό προσφέρει, πέρα από τη GPU, ενσωματωμένα εργαλεία για την οργάνωση των πειραμάτων, τη διαχείριση των δεδομένων και την καταγραφή των αποτελεσμάτων, επιτρέποντας μια ολιστική προσέγγιση στην ανάπτυξη του έργου.

- **Software**

Όσον αφορά το λογισμικό, η γλώσσα προγραμματισμού που χρησιμοποιήθηκε ήταν η Python, η οποία αποτελεί το πρότυπο για την ανάπτυξη εφαρμογών μηχανικής και βαθιάς μάθησης. Η Python παρέχει ένα ευρύ φάσμα βιβλιοθηκών που διευκολύνουν την υλοποίηση και εκπαίδευση νευρωνικών δικτύων. Η πιο βασική βιβλιοθήκη είναι η **PyTorch** [31] για την ανάπτυξη και εκπαίδευση του νευρωνικού δικτύου, καθώς και άλλες βιβλιοθήκες όπως οι **NumPy** [32], **Matplotlib** [33] και **Timm** για την επεξεργασία δεδομένων, την οπτικοποίηση αποτελεσμάτων και τη φόρτωση προεκπαιδευμένων μοντέλων.

4. Πειραματικά Αποτελέσματα

4.1 Εισαγωγή

Ο στόχος του παρόντος κεφαλαίου είναι η αξιολόγηση της απόδοσης του μοντέλου στα τρία διαφορετικά σύνολα δεδομένων, τα οποία αναλύθηκαν στο προηγούμενο κεφάλαιο, το καθένα από τα οποία εισάγει διαφορετικές προκλήσεις στην ανίχνευση deepfake. Η ανάλυση επικεντρώνεται όχι μόνο στη συνολική ακρίβεια ταξινόμησης αλλά και σε κρίσιμες μετρικές, όπως η ευαισθησία (recall), η εξειδίκευση (specificity) και η τιμή του AUC, με σκοπό να αξιολογηθεί η ικανότητα του μοντέλου να διαχειρίζεται ποικίλες μορφές παραποίησης.

Η αξιολόγηση πραγματοποιείται μέσα από τη σύγκριση διαφορετικών αρχιτεκτονικών βαθιάς μάθησης, με το EfficientNet-B0 να αποτελεί την κύρια επιλογή λόγω της αποδοτικότητάς του, ενώ παράλληλα εξετάζεται η απόδοσή του σε σχέση με το βασικό μοντέλο (baseline model). Επιπλέον, η διερεύνηση της υπερπροσαρμογής και της σταθερότητας του μοντέλου παρέχει χρήσιμες πληροφορίες για την ικανότητά του να γενικεύει πέρα από τα δεδομένα εκπαίδευσης. Η ενότητα αυτή περιλαμβάνει την ανάλυση των μετρικών απόδοσης, την οπτικοποίηση των καμπυλών μάθησης, καθώς και την παρουσίαση των πινάκων σύγκρισης. Τα αποτελέσματα της μελέτης συμβάλλουν στην αξιολόγηση των υπαρχουσών μεθόδων ανίχνευσης deepfake, ενώ παράλληλα προτείνονται πιθανές βελτιώσεις που θα μπορούσαν να ενισχύσουν την ακρίβεια και τη γενίκευση του μοντέλου.

4.2 Εκπαίδευση του μοντέλου και Επικύρωση

Ρυθμίσεις εκπαίδευσης

- **Βελτιστοποιητής:** Χρησιμοποιήθηκε ο Adam optimizer με ρυθμό μάθησης (learning rate) ίσο με 0.0001 και βάρος κανονικοποίησης (weight decay) 1e-3. Ο ρυθμός μάθησης καθορίζει πόσο γρήγορα ή αργά το μοντέλο θα προσαρμόσει τα βάρη του κατά τη διάρκεια της εκπαίδευσης. Ένας πολύ μικρός ρυθμός μάθησης μπορεί να κάνει την εκπαίδευση αργή, ενώ ένας πολύ μεγάλος μπορεί να οδηγήσει σε μη συγκλίνουσες ή υπερβολικά ευαίσθητες προσαρμογές. Για το συγκεκριμένο πρόβλημα, ο ρυθμός μάθησης 0.0001 επιλέχθηκε για να εξασφαλίσει σταδιακή και ελεγχόμενη εκπαίδευση, αποφεύγοντας τις υπερβολικές μεταβολές.

- **Αριθμός εποχών**: Η εκπαίδευση πραγματοποιήθηκε για 10 εποχές (epochs), με το μοντέλο να επεξεργάζεται ολόκληρο το σύνολο εκπαίδευσης σε κάθε εποχή. Ο αριθμός των epochs είναι συνήθως θέμα πειραματισμού, 10 epochs είναι ένα κοινό σημείο εκκίνησης για προβλήματα ταξινόμησης.

- **Συνάρτηση κόστους:** Έχει χρησιμοποιηθεί η συνάρτηση **BCEWithLogitsLoss** που συνδυάζει το Binary Cross Entropy loss με τη συνάρτηση ενεργοποίησης sigmoid. Αυτή η συνάρτηση είναι κατάλληλη για προβλήματα δυαδικής ταξινόμησης, όπως στην περίπτωση του deepfake όπου έχουμε δύο κατηγορίες, μία πραγματική και μία ψεύτικη εικόνα.

```
epochs = 10
lr = 0.0001

torch.manual_seed(SEED)
Net = Network(base_model_name="efficientnet_b0")
Net.unfreeze_last_layers(5)  # Unfreeze the last 5 layers
model = Net.to(device)
optimizer = Adam(model.parameters(), lr=lr, weight_decay=1e-3)
loss_fn = nn.BCEWithLogitsLoss()
```

Πίνακας 4.1: Ρυθμίσεις εκπαίδευσης

4.2.1 Διαδικασία Εκπαίδευσης

Η διαδικασία εκπαίδευσης του μοντέλου βασίστηκε σε επαναληπτικούς βρόχους επεξεργασίας δεδομένων από το σύνολο εκπαίδευσης με στόχο τη βελτιστοποίηση των βαρών. Η διαδικασία περιλαμβάνει τα εξής βήματα:

Αρχικά, η εκπαίδευση πραγματοποιείται μέσω της επεξεργασίας του συνόλου δεδομένων, το οποίο χωρίζεται σε μικρότερα υποσύνολα (batches) για αποτελεσματικότερη διαχείριση της μνήμης και επιτάχυνση της διαδικασίας. Για κάθε παρτίδα, τα δεδομένα και οι ετικέτες φορτώνονται και μεταφέρονται στη συσκευή εκτέλεσης (GPU ή CPU). Στη συνέχεια, το μοντέλο υπολογίζει την έξοδο για τα δεδομένα εισόδου μέσω της διαδικασίας προς τα εμπρός πέρασμα (forward pass). Έπειτα, υπολογίζεται η απώλεια με τη χρήση της συνάρτησης **BCEWithLogitsLoss**. Τέλος, με την διαδικασία της οπισθοδιάδοσης ενημερώνονται οι παράμετροι του μοντέλου μέσω του βελτιστοποιητή Adam. Στο τέλος κάθε epoch, εκτυπώνονται ο μέσος όρος της απώλειας και το ποσοστό ακρίβειας.

```
for batch_idx, (data, target, _) in enumerate(train_loader):
        data,           targets          =          data.to(device),
target.float().unsqueeze(1).to(device)
        optimizer.zero_grad()


        output = model(data)
        loss = loss_fn(output, targets)
        loss.backward()
        optimizer.step()
        total_loss += loss.item()
        preds = torch.round(torch.sigmoid(output))
        corrects += (preds == targets).sum().item()


avg_loss = total_loss / len(train_loader.dataset)
accuracy = 100. * corrects / len(train_loader.dataset)
train_losses.append(avg_loss)
```

Πίνακας 4.2: Βρόγχος εκπαίδευσης.

4.2.2 Επικύρωση

Η διαδικασία επικύρωσης πραγματοποιείται στο τέλος κάθε επανάληψης της εκπαίδευσης και περιλαμβάνει την αξιολόγηση του μοντέλου με τη χρήση δεδομένων ελέγχου. Όπως προαναφέρθηκε, τα δεδομένα αυτά είναι ανεξάρτητα από τα δεδομένα εκπαίδευσης, και δεν συμμετέχουν στη βελτιστοποίηση των βαρών του μοντέλου.

Αρχικά, με τη χρήση της συνάρτησης **model.eval** το μοντέλο τίθεται σε κατάσταση αξιολόγησης, απενεργοποιώντας όλες τις τεχνικές κανονικοποίησης ώστε οι προβλέψεις να είναι συνεπείς. Στη συνέχεια, τα δεδομένα επικύρωσης φορτώνονται σε παρτίδες και επεξεργάζονται από το μοντέλο χωρίς ενημέρωση των βαρών, δηλαδή χωρίς οπισθοδιάδοση. Τελικά, για κάθε παρτίδα υπολογίζεται η απώλεια ενώ οι προβλέψεις αποθηκεύονται για περαιτέρω ανάλυση.

```
 model.eval()
val_loss, corrects = 0,
0 all_preds = []
all_probs = []
all_targets = []
with torch.no_grad():
  for data, target, _ in val_loader:
    data, targets = data.to(device),
    target.float().unsqueeze(1).to(device)

    output = model(data)
    loss = loss_fn(output,
    targets) val_loss +=
    loss.item()
    preds =
    torch.round(torch.sigmoid(output))
    corrects += (preds ==
    targets).sum().item()

    all_preds.extend(preds.cpu().numpy())
    all_probs.extend(torch.sigmoid(output).cpu().numpy())
    all_targets.extend(targets.cpu().numpy())

avg_val_loss = val_loss /
len(val_loader.dataset) val_accuracy = 100. *
corrects / len(val_loader.dataset)
val_accuracies.append(val_accuracy)
```

Πίνακας 4.3: Βρογχος επικύρωσης.

Κατά την ολοκλήρωση της επεξεργασίας όλων των δεδομένων επικύρωσης υπολογίζονται οι παρακάτω μετρικές και δείκτες για την κατανόηση της απόδοσης του μοντέλου.

AUC (Area Under the ROC Curve): Υποδεικνύει την ικανότητα του μοντέλου να διαχωρίζει τις δύο κατηγορίες (real/fake). Τιμές κοντά στο 1 δείχνουν εξαιρετική απόδοση.

Accuracy (Ακρίβεια): Το ποσοστό των σωστών προβλέψεων σε σχέση με το συνολικό πλήθος παραδειγμάτων.

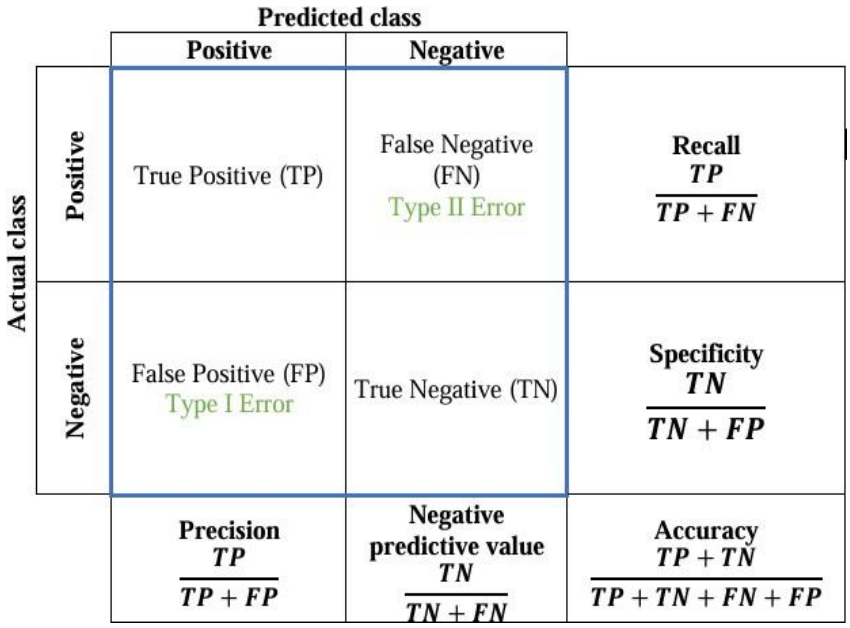
Specificity (Εξειδίκευση): Η ικανότητα του μοντέλου να αναγνωρίζει σωστά τις πραγματικές εικόνες.

Precision (Ακρίβεια Προβλέψεων): Το ποσοστό των σωστών deepfake προβλέψεων σε σχέση με όλες τις θετικές προβλέψεις.

Recall (Ανάκληση): Η αναλογία των σωστών deepfake προβλέψεων σε σχέση με όλα τα πραγματικά deepfake δείγματα.

```
# ------------ Final Evaluation Metrics -
---
 auc = roc_auc_score(all_targets,
all_probs)
tn, fp, fn, tp = confusion_matrix(all_targets,
all_preds).ravel() accuracy = (tp + tn) / (tp +
tn + fp + fn)
sensitivity = tp / (tp + fn)
specificity = tn / (tn + fp)
precision = tp / (tp + fp)
```

Πίνακας 4.4: Υπολογισμός μετρικών αξιολόγησης.



Εικόνα 4.1: Απεικόνιση πίνακα σύγχυσης.

4.3 Διαδικασία Πρόωρης Διακοπής Εκπαίδευσης

Η διαδικασία πρόωρης διακοπής εκπαίδευσης (early stopping) αποτελεί μία από τις τεχνικές κανονικοποίησης, η οποία χρησιμοποιήθηκε ως στρατηγική για την αποτροπή της υπερπροσαρμογής (overfitting). Αρχικά, παρακολουθείται η τιμή της απώλειας στο σύνολο επικύρωσης σε κάθε εποχή. Εφόσον, η απώλεια επικύρωσης δεν παρουσιάζει βελτίωση για 3 συνεχόμενες επαναλήψεις (patience), η εκπαίδευση τερματίζεται πρόωρα και αποθηκεύεται το καλύτερο μοντέλο. Ο σκοπός της χρήσης του είναι να εξοικονομεί χρόνο και πόρους, διασφαλίζοντας ότι το μοντέλο δεν εκπαιδεύεται πέρα από το βέλτιστο σημείο.

```
# Early Stopping Parameters
patience = 3 # Number of epochs to wait for
improvement best_val_loss = float('inf')
trigger = 0

# Early Stopping
      if   avg_val_loss   <
   best_val_loss:
   best_val_loss         =
   avg_val_loss  trigger  =
   0
   torch.save(model.state_dict(), 'best_model.pth') #
   Save best model print("Best model saved.")
  else:
   trigger += 1
   print(f"Early stopping trigger:
   {trigger}/{patience}") if trigger >=
   patience:
     print("Early stopping
     activated.") break
```

Πίνακας 4.6: Μηχανισμός early stopping για τη σταθεροποίηση της απόδοσης του νευρωνικού δικτύου.

4.3.1 Ανάλυση της Επίδρασης του Early Stopping και των καμπυλών μάθησης

Η τεχνική του early stopping εφαρμόστηκε στα τρία διαφορετικά σύνολα δεδομένων. Τα αποτελέσματα δείχνουν ότι το μοντέλο σταμάτησε την εκπαίδευση σε διαφορετικά χρονικά σημεία ανάλογα με τη συμπεριφορά της απώλειας επικύρωσης, επιτρέποντας την επιλογή της βέλτιστης εκδοχής του. Επιπρόσθετα, η αποτελεσματικότητα αυτής της στρατηγικής αξιολογήθηκε μέσω των καμπυλών μάθησης, οι οποίες αποτυπώνουν την συμπεριφορά του μοντέλου σε κάθε εποχή εκπαίδευσης. Τα διαγράμματα που ακολουθούν απεικονίζουν τη σχέση μεταξύ της απώλειας εκπαίδευσης και της ακρίβειας επικύρωσης, παρέχοντας πληροφορίες για την κατανόηση της διαδικασίας εκπαίδευσης και πιθανών προβλημάτων υπερπροσαρμογής.

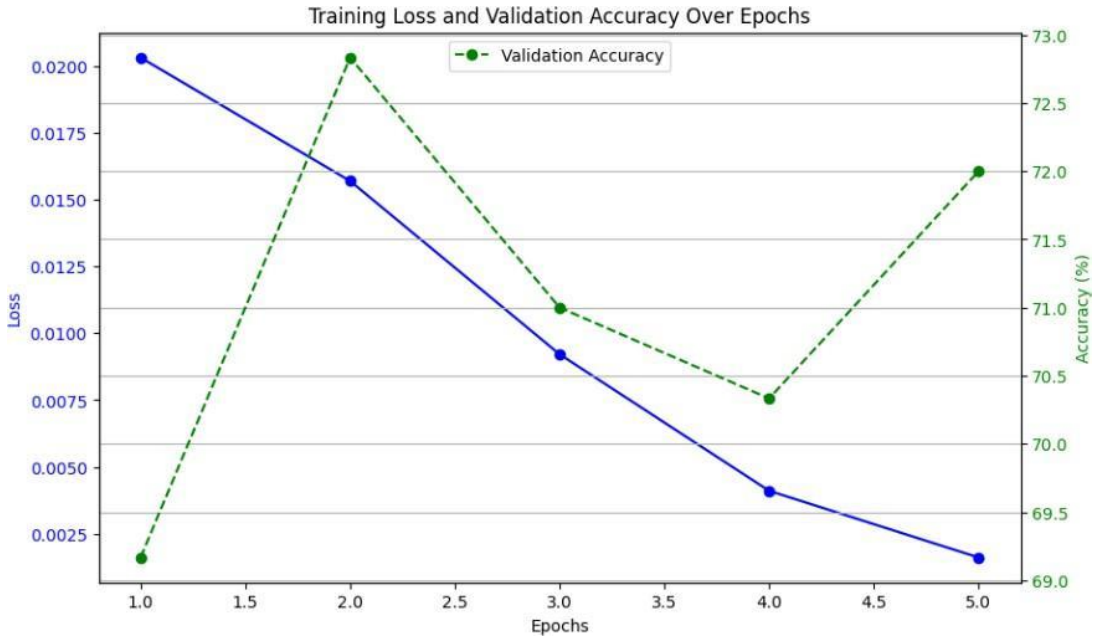
Αποτελέσματα

DFDC
Epoch 1/10 ---------- Train Loss: 0.0203 train Acc: 61.6250% Validation Loss: 0.0189 Validation Accuracy: 68.33% Best model saved. Epoch 2/10 ---------- Train Loss: 0.0157 train Acc: 77.0417% Validation Loss: 0.0178 Validation Accuracy: 72.50% Best model saved. Epoch 3/10 - Train Loss: 0.0092 train Acc: 88.4167% Validation Loss: 0.0198 Validation Accuracy: 71.00% Early stopping trigger: 1/3 Epoch 4/10 ---------- Train Loss: 0.0042 train Acc: 95.0000% Validation Loss: 0.0247 Validation Accuracy: 69.00% Early stopping trigger: 2/3 Epoch 5/10 ---------- Train Loss: 0.0017 train Acc: 98.9583% Validation Loss: 0.0260 Validation Accuracy: 69.33% Early stopping trigger: 3/3 Early stopping activated.

DFMNIST+
Epoch 1/10 ---------- Train Loss: 0.0184 train Acc: 71.2395% Validation Loss: 0.0242 Validation Accuracy: 38.46% Best model saved. Epoch 2/10 ---------- Train Loss: 0.0104 train Acc: 87.1239% Validation Loss: 0.0187 Validation Accuracy: 76.44% Best model saved. Epoch 3/10 ---------- Train Loss: 0.0049 train Acc: 95.7882% Validation Loss: 0.0118 Validation Accuracy: 86.06% Best model saved. Epoch 4/10 ---------- Train Loss: 0.0017 train Acc: 99.3983% Validation Loss: 0.0122 Validation Accuracy: 85.58% Early stopping trigger: 1/3 Epoch 5/10 ---------- Train Loss: 0.0009 train Acc: 100.0000% Validation Loss: 0.0134 Validation Accuracy: 86.06% Early stopping trigger: 2/3 Epoch 6/10 ---------- Train Loss: 0.0005 train Acc: 99.8797% Validation Loss: 0.0105 Validation Accuracy: 88.94% Best model saved. Epoch 7/10 ---------- Train Loss: 0.0003 train Acc: 99.8797% Validation Loss: 0.0100 Validation Accuracy: 91.35% Best model saved. Epoch 8/10 ---------- Train Loss: 0.0003 train Acc: 99.8797% Validation Loss: 0.0148 Validation Accuracy: 86.06% Early stopping trigger: 1/3 Epoch 9/10 ---------- Train Loss: 0.0002 train Acc: 100.0000% Validation Loss: 0.0095 Validation Accuracy: 92.79% Best model saved. Epoch 10/10 ---------- Train Loss: 0.0001 train Acc: 100.0000% Validation Loss: 0.0103 Validation Accuracy: 91.83% Early stopping trigger: 1/3

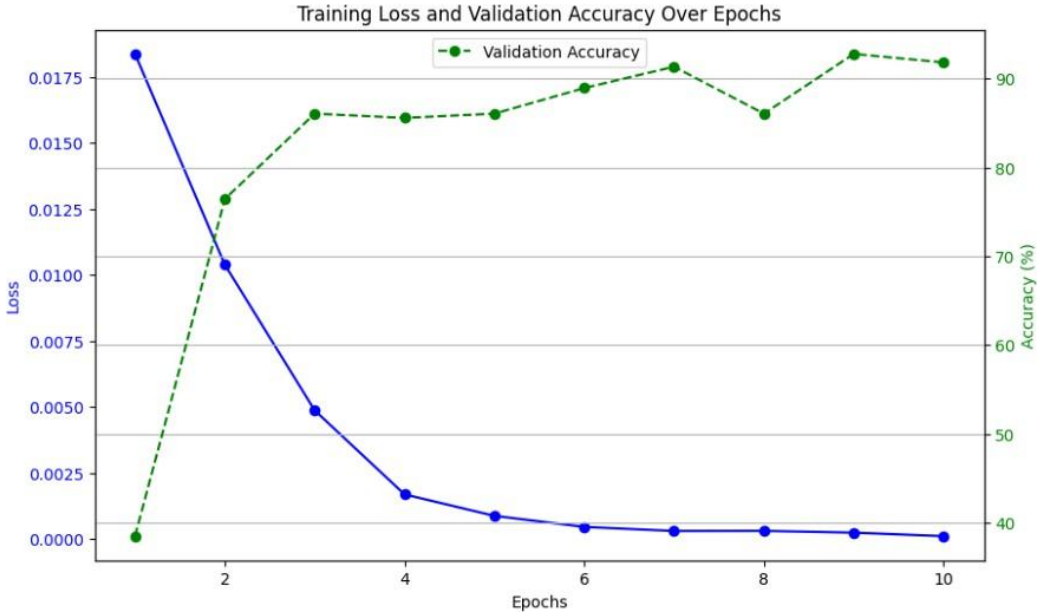
EXP
Epoch 1/10 ---------- Train Loss: 0.0207 train Acc: 60.2941% Validation Loss: 0.0191 Validation Accuracy: 69.19% Best model saved. Epoch 2/10 ---------- Train Loss: 0.0145 train Acc: 79.5956% Validation Loss: 0.0162 Validation Accuracy: 76.28% Best model saved. Epoch 3/10 ---------- Train Loss: 0.0071 train Acc: 92.3407% Validation Loss: 0.0149 Validation Accuracy: 78.73% Best model saved. Epoch 4/10 ---------- Train Loss: 0.0029 train Acc: 97.7328% Validation Loss: 0.0165 Validation Accuracy: 82.15% Early stopping trigger: 1/3 Epoch 5/10 ---------- Train Loss: 0.0015 train Acc: 98.7132% Validation Loss: 0.0185 Validation Accuracy: 81.17% Early stopping trigger: 2/3 Epoch 6/10 ---------- Train Loss: 0.0007 train Acc: 99.5711% Validation Loss: 0.0174 Validation Accuracy: 81.42% Early stopping trigger: 3/3 Early stopping activated.

Στο **DFDC** σύνολο δεδομένων, η απώλεια εκπαίδευσης μειώνεται σταθερά (Σχήμα 4.2), γεγονός που δείχνει ότι το μοντέλο προσαρμόζεται στα δεδομένα εκπαίδευσης. Το early stopping ενεργοποιήθηκε στην 5η εποχή, καθώς η απώλεια επικύρωσης εμφάνισε τη χαμηλότερη τιμή της στη 2η εποχή (0.0178) και στη συνέχεια αυξήθηκε. Παρόλο που η ακρίβεια εκπαίδευσης αυξήθηκε σημαντικά, η συνεχής μείωση της απώλειας εκπαίδευσης σε συνδυασμό με την αύξηση της απώλειας επικύρωσης δείχνει ότι το μοντέλο άρχισε να απομνημονεύει τα δεδομένα εκπαίδευσης οδηγώντας σε υπερπροσαρμογή. Για το συγκεκριμένο σύνολο δεδομένων η ενεργοποίηση του early stopping αποδείχθηκε χρήσιμη καθώς η συνέχιση της εκπαίδευσης πιθανότατα θα επιδείνωνε τη γενίκευση του μοντέλου.



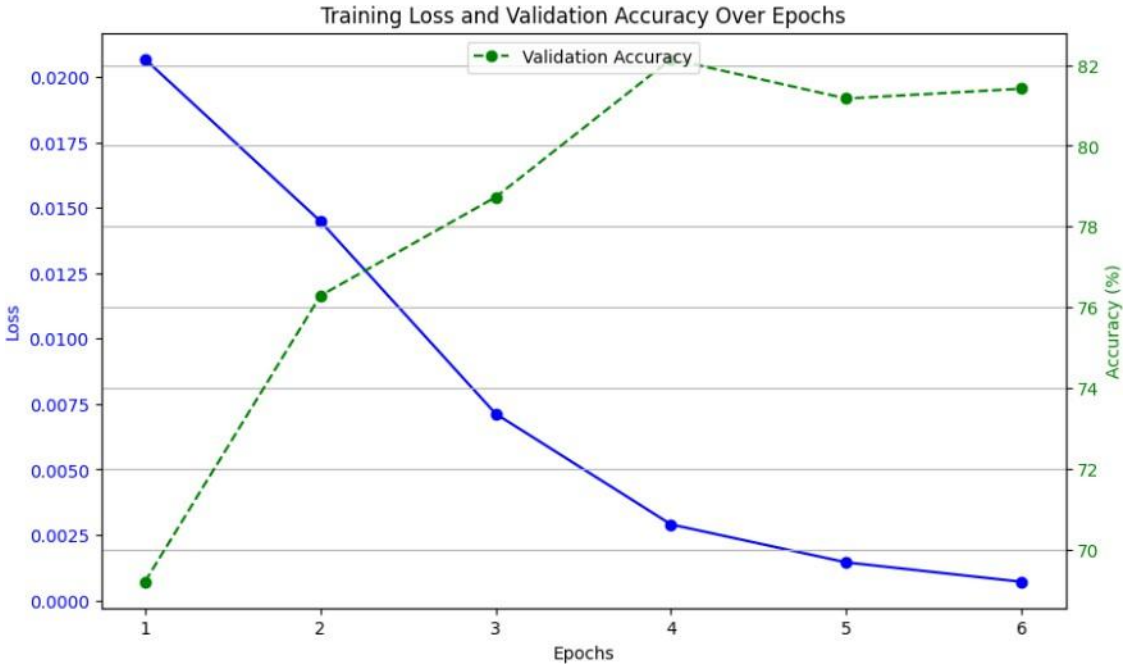
Σχήμα 4.2: Εξέλιξη της απώλειας εκπαίδευσης και της ακρίβειας επικύρωσης κατά τη διάρκεια της εκπαίδευσης του μοντέλου για το σύνολο δεδομένων DFDC.

Αντίθετα, στο **DFMNIST+** σύνολο δεδομένων, η απώλεια εκπαίδευσης μειώνεται σταθερά ενώ η ακρίβεια επικύρωσης αυξάνεται προοδευτικά μέχρι την $9^{η}$ εποχή όπου σταθεροποιείται (Εικόνα 4.3). Η απώλεια επικύρωσης διατηρείται σε χαμηλά επίπεδα, όπου η χαμηλότερη τιμή καταγράφεται στο 0.0095, χωρίς έντονες αυξομειώσεις, γεγονός που δείχνει ότι το μοντέλο δεν εμφάνισε υπερπροσαρμογή. Σε αντίθεση με το DFDC, οι δύο καμπύλες παραμένουν παράλληλες, επιτρέποντας την εκπαίδευση χωρίς απώλεια γενίκευσης. Επιπλέον, το early stopping δεν ενεργοποιήθηκε σε πρόωρο στάδιο καθώς η απώλεια επικύρωσης δεν παρουσίασε απότομες αυξήσεις και το μοντέλο διατήρησε υψηλή ακρίβεια ακόμα και μετά τη 10η εποχή.



Σχήμα 4.3: Εξέλιξη της απώλειας εκπαίδευσης και της ακρίβειας επικύρωσης κατά τη διάρκεια της εκπαίδευσης του μοντέλου για το σύνολο δεδομένων DFMNIST+.

Στο **EXP** σύνολο δεδομένων, η ακρίβεια επικύρωσης αυξάνεται μέχρι την 3η εποχή, ενώ η απώλεια επικύρωσης μειώνεται (Εικόνα 4.4). Το early stopping ενεργοποιήθηκε στην 6η εποχή, καθώς η απώλεια επικύρωσης σταμάτησε να μειώνεται μετά την 3η εποχή (0.0149), όπου καταγράφηκε η χαμηλότερη τιμή της, και παρουσίασε διακυμάνσεις. Παρόλο που η ακρίβεια επικύρωσης συνέχισε να βελτιώνεται, η γενική συμπεριφορά της απώλειας επικύρωσης έδειξε ότι το μοντέλο είχε φτάσει στο βέλτιστο σημείο του νωρίτερα, καθιστώντας την περαιτέρω εκπαίδευση μη απαραίτητη.



Σχήμα 4.4: Εξέλιξη της απώλειας εκπαίδευσης και της ακρίβειας επικύρωσης κατά τη διάρκεια της εκπαίδευσης του μοντέλου για το σύνολο δεδομένων EXP.

Τελικά, τα αποτελέσματα επιβεβαιώνουν ότι το early stopping ήταν ιδιαίτερα αποτελεσματικό διασφαλίζοντας ότι η εκπαίδευση σταμάτησε στο σωστό σημείο. Αντίθετα, στο δεύτερο σύνολο δεδομένων, η χρήση της πρόωρης διακοπής εκπαίδευσης δεν είχε τόσο έντονη επίδραση. Επιπλέον, επιβεβαιώνεται ότι η αποτελεσματικότητα αυτής της τεχνικής εξαρτάται από τη φύση του συνόλου δεδομένων καθώς οι διαφορετικές αρχιτεκτονικές deepfake εικόνων μπορεί να επηρεάσουν την ικανότητα γενίκευσης του μοντέλου.

4.4 Συγκριτική Αξιολόγηση των Μοντέλων

Για τη συγκριτική αξιολόγηση των μοντέλων, είναι κρίσιμο να διασφαλιστεί ότι οι συνθήκες εκπαίδευσης παραμένουν σταθερές, ώστε τα αποτελέσματα να είναι άμεσα συγκρίσιμα. Συνεπώς, όλα τα μοντέλα εκπαιδεύτηκαν και αξιολογήθηκαν υπό τις ίδιες υπερ-παραμέτρους, διατηρώντας σταθερό τον αριθμό των εποχών, τον ρυθμό μάθησης (learning rate) και το μέγεθος του batch (batch size). Επιπλέον, η αρχικοποίηση των βαρών παρέμεινε σταθερή διασφαλίζοντας την αναπαραγωγιμότητα των αποτελεσμάτων. Η τήρηση αυτών των σταθερών παραμέτρων εγγυάται ότι οι διαφορές στην απόδοση των μοντέλων αποδίδονται αποκλειστικά στις διαφορές των αρχιτεκτονικών τους και όχι σε εξωτερικούς παράγοντες που θα μπορούσαν να επηρεάσουν τη διαδικασία μάθησης.

Το **ResNet-18** αποτελεί την κύρια αρχιτεκτονική που χρησιμοποιήθηκε για το βασικό μοντέλο (baseline model). Προτάθηκε από τους Kaiming He και τους συνεργάτες του και είναι ένα βαθύ συνελικτικό νευρωνικό δίκτυο που ανήκει στην οικογένεια των Residual Networks (ResNets). Οι ResNets εισήγαγαν τη λογική της υπολειμματικής μάθησης για να αντιμετωπίσουν το πρόβλημα της υποβάθμισης της απόδοσης σε πολύ βαθιά νευρωνικά δίκτυα. Το **ResNet-18** είναι μία από τις πιο ελαφριές εκδόσεις αυτής της αρχιτεκτονικής, με 18 στρώματα, και έχει σχεδιαστεί ώστε να παρέχει αποτελεσματική εκπαίδευση και βελτιωμένη γενίκευση. Αυτή η προσέγγιση έχει αποδειχθεί ιδιαίτερα επιτυχημένη σε πληθώρα εφαρμογών αναγνώρισης εικόνας, όπως ο διαγωνισμός **ILSVRC 2015**, ξεπερνώντας αρχιτεκτονικές όπως το **VGG**, βελτιώνοντας την αποδοτικότητα. [34]

Dataset	Model	Accuracy	AUC	Sensitivity(Recall)	Specificity	Precision
DFDC	**EfficientNet**	**0.7200**	0.7916	0.8252	0.6082	0.6911
	Baseline Model	0.6466	0.7070	0.7152	0.5739	0.6406
DFMNIST+	**EfficientNet**	**0.9183**	0.9611	0.9453	0.8750	0.9237
	Baseline Model	0.7355	0.8218	0.9609	0.3750	0.7110
EXP	**EfficientNet**	**0.8142**	0.8935	0.8214	0.8075	0.7970
	Baseline Model	0.6552	0.7126	0.5714	0.7324	0.6627

Πίνακας 4.7: Σύγκριση συνολικών αποτελεσμάτων.

Ανάλυση επίδοσης των μοντέλων

Η συγκριτική αξιολόγηση των μοντέλων αναδεικνύει τη σημασία της επιλογής της αρχιτεκτονικής και της προσαρμογής της στις ιδιαιτερότητες κάθε συνόλου δεδομένων. Η ανάλυση βασίζεται στις μετρικές απόδοσης που καταγράφονται στον πίνακα, οι οποίες περιλαμβάνουν την ακρίβεια (accuracy), την περιοχή κάτω από την καμπύλη ROC (AUC), την ευαισθησία/ανάκληση (sensitivity/recall), την εξειδίκευση (specificity) και την ακρίβεια ταξινόμησης (precision). Αυτές οι μετρικές παρέχουν μια πλήρη εικόνα της ικανότητας των μοντέλων να ανιχνεύουν deepfake εικόνες, ενώ επιτρέπουν τη σύγκριση μεταξύ διαφορετικών συνόλων δεδομένων και αρχιτεκτονικών. Όπως φαίνεται από τον πίνακα το EfficientNet υπερτερεί σταθερά έναντι του βασικού μοντέλου (baseline model) σε όλα τα σύνολα δεδομένων, παρουσιάζοντας υψηλότερη ακρίβεια. Παρακάτω αναλύονται όλες οι μετρικές για καθένα από τα τρία σύνολα δεδομένων.

Στο **DFDC** σύνολο δεδομένων, το EfficientNet κατέγραψε ακρίβεια 72.00% και AUC 0.7916, επιτυγχάνοντας σημαντική αύξηση έναντι του βασικού μοντέλου, το οποίο σημείωσε ακρίβεια 64.66% και AUC 0.7070. Αυτή η διαφορά υποδηλώνει ότι το EfficientNet έχει καλύτερη ικανότητα κατηγοριοποίησης των εικόνων, ανιχνεύοντας με μεγαλύτερη επιτυχία τις deepfake εικόνες. Ωστόσο, παρατηρείται ανισορροπία μεταξύ ευαισθησίας (0.8252) και εξειδίκευσης (0.6082). Το γεγονός ότι η ευαισθησία είναι υψηλή σημαίνει πως το μοντέλο ανιχνεύει με μεγάλη επιτυχία τις deepfake εικόνες, αλλά ταυτόχρονα η χαμηλή εξειδίκευση δείχνει ότι ταξινομεί λανθασμένα πολλές πραγματικές εικόνες ως deepfake. Αυτή η ασυμμετρία μπορεί να επηρεάσει την αξιοπιστία της ανίχνευσης, καθώς το ιδανικό μοντέλο απαιτεί ισορροπία μεταξύ των δύο μετρικών.

Αντίθετα, το βασικό μοντέλο παρουσίασε πιο κοντινές τιμές μεταξύ ευαισθησίας (0.7152) και εξειδίκευσης (0.5739). Αν και η συνολική του απόδοση είναι κατώτερη, η μικρότερη διαφορά μεταξύ των δύο μετρικών δείχνει ότι η συμπεριφορά του μοντέλου είναι πιο ισορροπημένη.

Η έλλειψη αυτής της ισορροπίας στο EfficientNet πιθανώς αποδίδεται στη μεγάλη ποικιλομορφία των deepfake τεχνικών που περιλαμβάνονται στο DFDC, οι οποίες μπορεί να οδήγησαν το μοντέλο σε υπερβολική ευαισθησία εις βάρος της εξειδίκευσης. Αυτό σημαίνει ότι η ταξινόμηση βασίζεται περισσότερο σε χαρακτηριστικά που εντοπίζονται στις ψεύτικες εικόνες, παρά στην ακριβή αναγνώριση των πραγματικών. Η εύρεση μεθόδων που βελτιώνουν τη γενίκευση του μοντέλου, ώστε να αυξηθεί η εξειδίκευση χωρίς να θυσιαστεί η ευαισθησία, αποτελεί σημαντική πρόκληση για την αποτελεσματική ανίχνευση deepfake εικόνων.

Στο **DFMNIST**+ σύνολο δεδομένων, το EfficientNet εμφάνισε την υψηλότερη συνολική ακρίβεια με ποσοστό 91.83% και AUC 0.9611, καταδεικνύοντας την ισχυρή ικανότητά του να διακρίνει deepfake από πραγματικές εικόνες. Σε αντίθεση με το DFDC, η ευαισθησία (0.9453) και η εξειδίκευση (0.8750) είναι αρκετά ισορροπημένες, γεγονός που δείχνει ότι το μοντέλο καταφέρνει να ανιχνεύσει τόσο τις ψεύτικες όσο και τις πραγματικές εικόνες με μεγάλη ακρίβεια. Αυτή η ισορροπία είναι ιδιαίτερα επιθυμητή, καθώς διασφαλίζει ότι το μοντέλο δεν εμφανίζει μεροληψία υπέρ μιας από τις δύο κατηγορίες, καθιστώντας το πιο αξιόπιστο για εφαρμογές ανίχνευσης deepfake.

Αντίθετα, το βασικό μοντέλο εμφάνισε σημαντικά χαμηλότερη συνολική ακρίβεια με ποσοστό 73.55% και AUC 0.8218, ενώ η ευαισθησία (0.9609) και η εξειδίκευση (0.3750) παρουσίασαν μεγάλη απόκλιση. Αυτό σημαίνει ότι το μοντέλο είχε την τάση να ανιχνεύει σχεδόν όλες τις deepfake εικόνες, αλλά απέτυχε σε μεγάλο βαθμό να ταξινομήσει σωστά τις πραγματικές εικόνες, αποδίδοντάς τους χαρακτηριστικά deepfake. Αυτή η υπερεστίαση στην ευαισθησία οδηγεί σε υψηλό αριθμός των πραγματικών εικόνων που ταξινομήθηκαν λανθασμένα ως ψεύτικες, γεγονός που περιορίζει την αξιοπιστία του συστήματος.

Η σημαντική βελτίωση που παρατηρείται στο EfficientNet πιθανώς οφείλεται στα χαρακτηριστικά του DFMNIST+, το οποίο επικεντρώνεται στις κινήσεις του προσώπου αντί για την απλή ανταλλαγή ταυτότητας. Η προσέγγιση αυτή φαίνεται να επιτρέπει στο μοντέλο να μάθει πιο σταθερά και αναγνωρίσιμα μοτίβα, βελτιώνοντας τη γενίκευση και οδηγώντας σε μια πιο ισορροπημένη ταξινόμηση. Η επιτυχία αυτή καταδεικνύει τη σημασία του σωστά επιλεγμένου συνόλου δεδομένου για την αποτελεσματική ανίχνευση deepfake εικόνων.

Στο **EXP** σύνολο δεδομένων, το EfficientNet παρουσίασε επίσης βελτιωμένη απόδοση, με ακρίβεια 81.42% και AUC 0.8935, υποδηλώνοντας ότι το μοντέλο κατάφερε να διακρίνει σε μεγάλο βαθμό της deepfake εικόνες από τις πραγματικές. Ενδιαφέρον παρουσιάζει το γεγονός ότι η ευαισθησία (0.8214) και η εξειδίκευση (0.8075) είναι σχεδόν ισορροπημένες, κάτι που σημαίνει ότι το μοντέλο αναγνωρίζει τόσο τις deepfake όσο και τις πραγματικές εικόνες με εξαιρετική ακρίβεια. Αυτή η ισορροπία αποτελεί το ζητούμενο σε ένα ταξινομητή, καθώς εξασφαλίζει ότι το σύστημα δεν είναι μεροληπτικό προς μία από τις δύο κατηγορίες, προσφέροντας αξιόπιστα αποτελέσματα ανίχνευσης. Αντίθετα, το βασικό μοντέλο παρουσίασε εμφανώς χαμηλότερη συνολική ακρίβεια 65.52% και AUC (0.7126), με σημαντική ανισορροπία μεταξύ ευαισθησίας (0.5714) και εξειδίκευσης (0.7324). Αυτό δείχνει ότι το βασικό μοντέλο είχε σαφώς μεγαλύτερη δυσκολία στη σωστή αναγνώριση των deepfake εικόνων, ταξινομώντας περισσότερες από αυτές λανθασμένα ως πραγματικές.

Η απόδοση του EfficientNet στο EXP σύνολο δεδομένων ενδέχεται να οφείλεται στο ότι οι deepfake εικόνες αυτού του συνόλου δεδομένων δημιουργήθηκαν με υψηλής ποιότητας τεχνικές επεξεργασίας, κάνοντας τη διάκριση μεταξύ αυθεντικών και ψεύτικων εικόνων πιο απαιτητική. Παρόλα αυτά, το γεγονός ότι η ευαισθησία και η εξειδίκευση βρίσκονται σε πολύ κοντινές τιμές δείχνει ότι το βελτιωμένο μοντέλο πέτυχε μια ιδιαίτερα σταθερή και ισορροπημένη ταξινόμηση, κάτι που το καθιστά πιο αξιόπιστο σε πραγματικές συνθήκες ανίχνευσης deepfake.

Τα αποτελέσματα καταδεικνύουν ότι η απόδοση του μοντέλου εξαρτάται από τη φύση των δεδομένων, με το DFMNIST+ να αποτελεί το καταλληλότερο σύνολο δεδομένων για την αρχιτεκτονική με το EfficientNet. Αυτό συμβαίνει επειδή παρέχει σαφέστερα χαρακτηριστικά για την εκπαίδευση, επιτρέποντας πιο σταθερή ταξινόμηση. Αντίθετα, το DFDC παρουσίασε προκλήσεις λόγω της ποικιλομορφίας των τεχνικών deepfake, ενώ το EXP απαιτούσε μεγαλύτερη ακρίβεια λόγω των χειροκίνητα επεξεργασμένων εικόνων.

Τέλος, το βασικό μοντέλο, το οποίο βασίζεται στο ResNet, εμφάνισε σε όλες τις περιπτώσεις χαμηλότερη απόδοση συγκριτικά με το EfficientNet, γεγονός που επιβεβαιώνει τη βελτιωμένη ικανότητα της τελευταίας αρχιτεκτονικής στην ανίχνευση deepfake εικόνων.

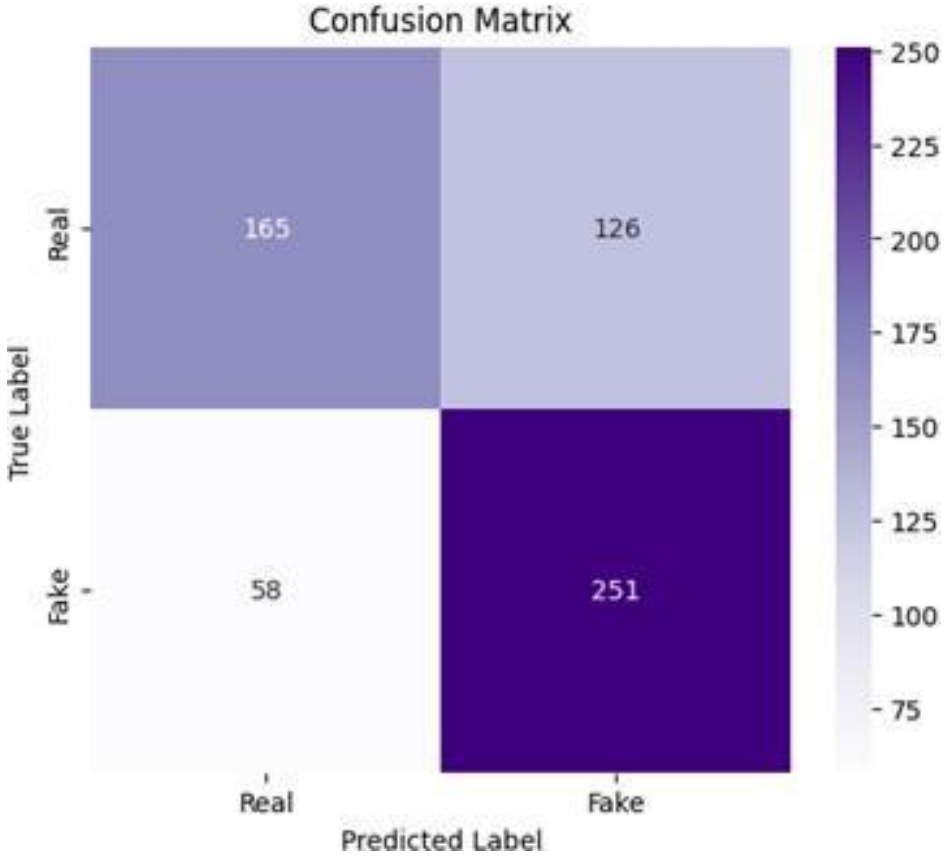
4.5 Ανάλυση Απόδοσης μέσω Πινάκων Σύγχυσης

Στην ενότητα αυτή, παρουσιάζονται οι πίνακες σύγχυσης (confusion matrices) για κάθε σύνολο δεδομένων, παρέχοντας μια οπτικοποιημένη ανάλυση της απόδοσης του μοντέλου. Οι θερμικοί χάρτες (heatmaps) απεικονίζουν τη συχνότητα σωστών και λανθασμένων ταξινομήσεων, επιτρέποντας την αξιολόγηση της ικανότητας του ταξινομητή να διακρίνει μεταξύ πραγματικών και παραποιημένων εικόνων. Η σύγκριση τους είναι απαραίτητη, καθώς επηρεάζει την αξιοπιστία της ανίχνευσης deepfake.

Κάθε πίνακας περιλαμβάνει τέσσερις βασικές τιμές:

- True Positive (TP): Ο αριθμός των ψεύτικων εικόνων που ταξινομήθηκαν σωστά ως ψεύτικες.
- True Negative (TN): Ο αριθμός των πραγματικών εικόνων που ταξινομήθηκαν σωστά ως πραγματικές.
- False Positive (FP): Ο αριθμός των πραγματικών εικόνων που ταξινομήθηκαν λανθασμένα ως ψεύτικες.
- False Negative (FN): Ο αριθμός των ψεύτικων εικόνων που ταξινομήθηκαν λανθασμένα ως πραγματικές.

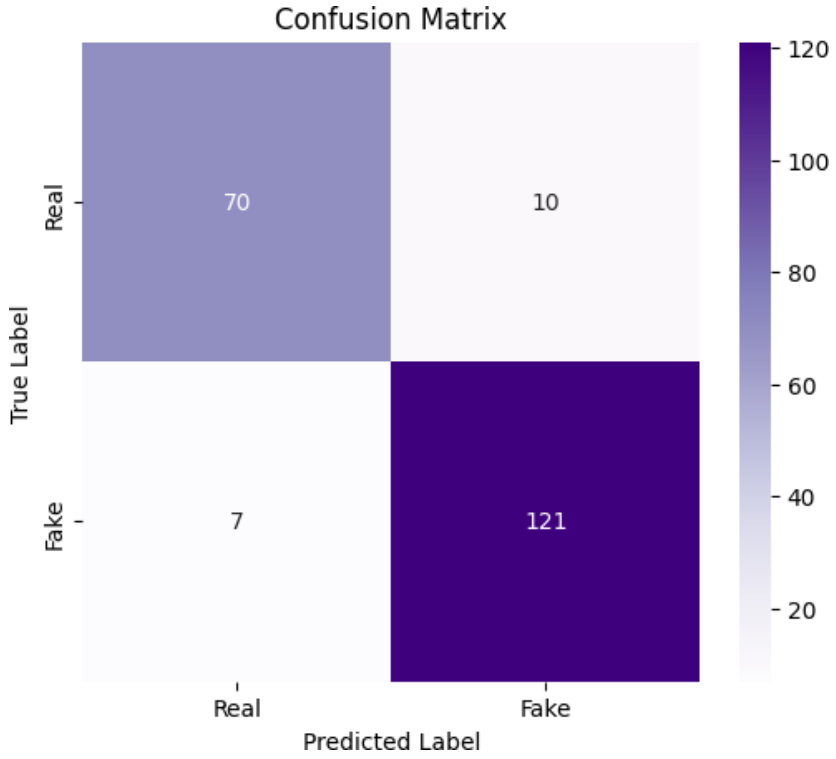
Στο **DFDC** σύνολο δεδομένων, ο πίνακας σύγχυσης (Σχήμα 4.5) δείχνει ότι το μοντέλο παρουσίασε καλή ικανότητα ανίχνευσης deepfake εικόνων. Συγκεκριμένα, καταγράφηκαν 251 TP και 165 TN, γεγονός που υποδηλώνει ότι το μεγαλύτερο ποσοστό των deepfake ταξινομήθηκε σωστά. Ωστόσο, οι τιμές FP ανέρχονται σε 126, μια τιμή αρκετά κοντινή με τα TN, αυτό δείχνει ότι το μοντέλο αντιμετώπισε δυσκολία στην διάκριση πραγματικών εικόνων. Πρακτικά, λίγο παραπάνω από τις μισές πραγματικές εικόνες ταξινομήθηκαν σωστά, ενώ οι υπόλοιπες αποδόθηκαν λανθασμένα στην κατηγορία των deepfake. Η FN τιμή των 58 υποδηλώνει ότι το μοντέλο διαθέτει υψηλή ευαισθησία (sensitivity), καθώς ανιχνεύει την πλειονότητα των ψεύτικων εικόνων. Παράλληλα όμως, η χαμηλότερη εξειδίκευση (specificity) καταδεικνύει την τάση του μοντέλου να ταξινομεί πραγματικές εικόνες ως deepfake, κάτι που έχει ήδη παρατηρηθεί και στην προηγουμένη ενότητα της αξιολόγησης.



Σχήμα 4.5: Πίνακας σύγχυσης του συνόλου δεδομένου DFDC.

Στο **DFMNIST+** σύνολο δεδομένων, ο πίνακας σύγχυσης (Σχήμα 4.6) δείχνει βελτιωμένη συνολική απόδοση, με σημαντικά λιγότερα σφάλματα. Συγκεκριμένα, καταγράφηκαν 121 TP και 70 TN υποδηλώνοντας πολύ μικρό αριθμό λανθασμένων προβλέψεων, ενώ οι τιμές FP και FN περιορίστηκαν στις 10 και 7 αντίστοιχα. Τα αποτελέσματα αυτά φανερώνουν ότι το μοντέλο διακρίνει με υψηλή ακρίβεια μεταξύ πλαστών και πραγματικών εικόνων.

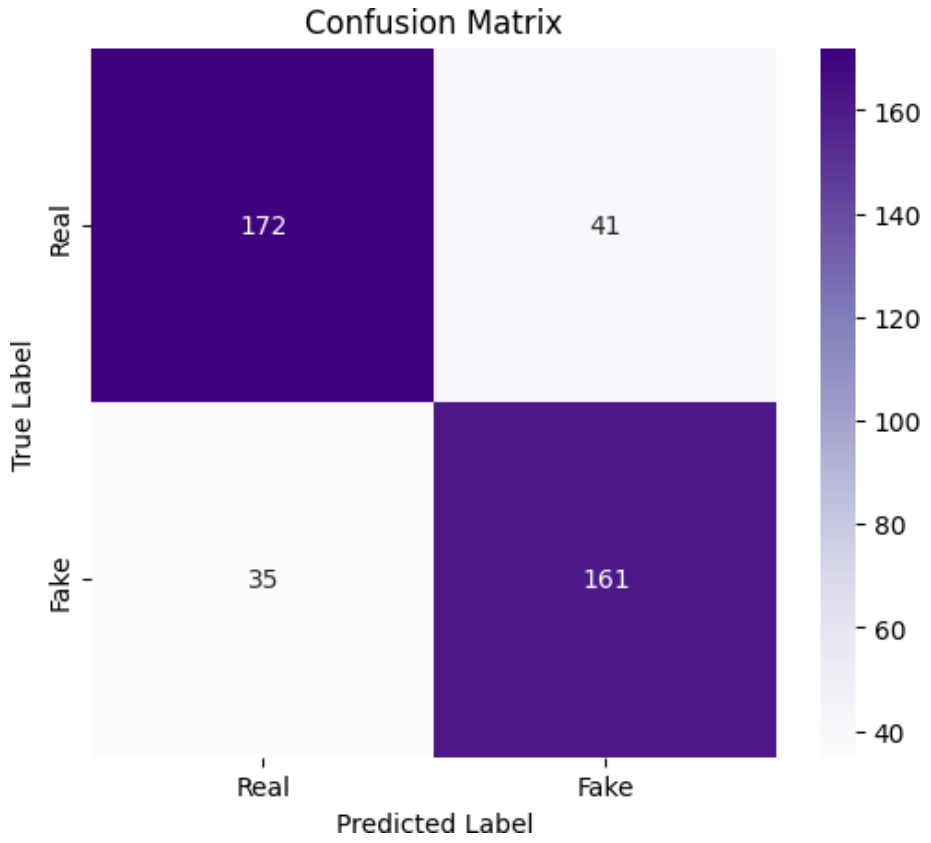
Η σχεδόν ισοδύναμη απόδοση στην αναγνώριση και των 2 κατηγοριών υποδηλώνει υψηλή εξειδίκευση (specificity) και ευαισθησία (sensitivity), οι οποίες βρίσκονται σε πολύ κοντινά επίπεδα. Αυτή η ισορροπία αποτελεί επιθυμητό χαρακτηριστικό για κάθε ταξινομητή, καθώς υποδεικνύει αξιόπιστη και συνεπής συμπεριφορά, χωρίς μεροληψία προς κάποια κατηγορία. Η απόδοση αυτή ενισχύει το συμπέρασμα που αναφέρθηκε στην προηγούμενη ενότητα, δηλαδή ότι το συγκεκριμένο σύνολο δεδομένων φαίνεται να παρέχει πιο καθαρά και διακριτά χαρακτηριστικά, διευκολύνοντας τη διαδικασία εκμάθησης.



Σχήμα 4.6: Πίνακας σύγχυσης του συνόλου δεδομένου DFMNIST+.

Στο **EXP** σύνολο δεδομένων, ο πίνακας σύγχυσης (Σχήμα 4.7) δείχνει ισορροπημένη απόδοση αλλά με μεγαλύτερο αριθμό λανθασμένων προβλέψεων σε σύγκριση με το DFMNIST+. Συγκεκριμένα, το μοντέλο ταξινόμησε σωστά 161 ψεύτικες εικόνες και 172 πραγματικές, ενώ έκανε 41 ψευδώς θετικές και 35 ψευδώς αρνητικές προβλέψεις. Οι ψευδώς θετικές και αρνητικές είναι αρκετά ισορροπημένες.

Αυτό σημαίνει ότι το μοντέλο κατάφερε να διακρίνει καλύτερα τις πραγματικές εικόνες, αλλά ταυτόχρονα ταξινόμησε ορισμένες deepfake εικόνες ως πραγματικές. Η απόδοσή του υποδηλώνει ότι διαχειρίζεται ικανοποιητικά την ισορροπία μεταξύ ευαισθησίας (sensitivity) και εξειδίκευσης (specificity). Ωστόσο, η ύπαρξη ψευδώς αρνητικών προβλέψεων (FN = 35) δείχνει ότι το μοντέλο μερικές φορές αποτυγχάνει να αναγνωρίσει deepfake εικόνες, γεγονός που μπορεί να επηρεάσει την αξιοπιστία του. Αυτή η συμπεριφορά ενδέχεται να σχετίζεται με τη φύση του EXP συνόλου δεδομένων, το οποίο πιθανώς περιλαμβάνει deepfake εικόνες με πιο διακριτά χαρακτηριστικά από τις αυθεντικές, καθιστώντας δυσκολότερη την ακριβή ταξινόμηση.



Σχήμα 4.7: Πίνακας σύγχυσης του συνόλου δεδομένου EXP.

Συμπερασματικά, η ανάλυση των πινάκων σύγχυσης ανέδειξε σημαντικές διαφορές στην απόδοση του μοντέλου ανάλογα με το σύνολο δεδομένων, επιβεβαιώνοντας τον καθοριστικό ρόλο της φύσης των δεδομένων στη διαδικασία ταξινόμησης. Στο DFDC σύνολο δεδομένων, η υψηλή ευαισθησία του μοντέλου επέτρεψε την επιτυχή ανίχνευση deepfake εικόνων, ωστόσο η χαμηλή εξειδίκευση οδήγησε σε λανθασμένες ταξινομήσεις πραγματικών εικόνων, πιθανώς λόγω της ποικιλομορφίας των deepfake τεχνικών που περιλαμβάνει. Αντίθετα, το DFMNIST+ παρουσίασε την καλύτερη ισορροπία μεταξύ ακρίβειας, εξειδίκευσης και ευαισθησίας, με χαμηλά ποσοστά σφαλμάτων, υποδηλώνοντας ότι η σαφήνεια των χαρακτηριστικών του διευκόλυνε την ταξινόμηση. Το EXP σύνολο δεδομένων ενώ διατήρησε ικανοποιητική απόδοση, το μοντέλο δυσκολεύτηκε περισσότερο στην αναγνώριση deepfake εικόνων, πιθανώς λόγω της υψηλής ποιότητας των επεξεργασμένων εικόνων του, που καθιστούν δυσκολότερη τη διάκριση μεταξύ αυθεντικών και παραποιημένων. Τελικά, το DFMNIST+ εμφάνισε την καλύτερη απόδοση, γεγονός που μπορεί να αποδοθεί στο ότι επικεντρώνεται στις κινούμενες εκφράσεις αντί για απλή ανταλλαγή ταυτότητας, βοηθώντας το μοντέλο να μάθει πιο σταθερά μοτίβα και να βελτιώσει τη γενίκευσή του. Τα αποτελέσματα αυτά υπογραμμίζουν τη σημασία της επιλογής και της προεπεξεργασίας των δεδομένων στην ανίχνευση deepfake, καθώς η ποιότητα και η δομή του συνόλου δεδομένων επηρεάζουν καθοριστικά την αποτελεσματικότητα του ταξινομητή.

4.6 Συμπεράσματα

Η παρούσα μελέτη ανέλυσε την απόδοση ενός ταξινομητή βαθιάς μάθησης για την ανίχνευση deepfake εικόνων, εστιάζοντας στη συγκριτική αξιολόγηση μεταξύ διαφορετικών συνόλων δεδομένων και αρχιτεκτονικών μοντέλων. Τα αποτελέσματα επιβεβαιώνουν ότι η απόδοση του μοντέλου εξαρτάται σε μεγάλο βαθμό από τη φύση των δεδομένων, τις τεχνικές δημιουργίας των deepfake εικόνων και τις επιλεγμένες αρχιτεκτονικές βαθιάς μάθησης.

Η εκπαίδευση του μοντέλου έδειξε ότι το EfficientNet-B0 υπερτερεί του βασικού μοντέλου (baseline model), το οποίο βασίζεται στο ResNet-18, επιτυγχάνοντας υψηλότερη ακρίβεια και ισορροπημένες μετρικές απόδοσης. Το EfficientNet-B0 απέδειξε την ικανότητά του να διαχειρίζεται διαφορετικές προκλήσεις, όπως η ποικιλομορφία των deepfake τεχνικών στο DFDC, οι κινούμενες εκφράσεις στο DFMNIST+ και οι εικόνες υψηλής ποιότητας του EXP.

Η χρήση της διακοπής εκπαίδευσης αποδείχθηκε ιδιαίτερα χρήσιμη για την αποφυγή της υπερπροσαρμογής καθώς το μοντέλο σταματά τη διαδικασία εκπαίδευσης στο βέλτιστο σημείο. Επίσης, οι καμπύλες μάθησης έδειξαν πως το μοντέλο παρουσίασε διαφορετική συμπεριφορά σε κάθε σύνολο εκπαίδευσης, με το DFMNIST+ να εμφανίζει την πιο σταθερή εκπαίδευση, χωρίς υπερπροσαρμογή.

Από τη συγκριτική αξιολόγηση των αποτελεσμάτων, προκύπτει ότι το DFMNIST+ αποτελεί το καταλληλότερο σύνολο δεδομένων για την EfficientNet-B0 αρχιτεκτονική, καθώς παρουσίασε την υψηλότερη ακρίβεια και τη μεγαλύτερη ισορροπία μεταξύ ευαισθησίας και εξειδίκευσης. Αντίθετα, το DFDC παρουσίασε προκλήσεις λόγω της ποικιλομορφίας των deepfake τεχνικών, ενώ το EXP απαιτούσε μεγαλύτερη ακρίβεια λόγω των χειροκίνητα επεξεργασμένων εικόνων.

Τέλος, το βασικό μοντέλο με την ResNet-18 αρχιτεκτονική εμφάνισε σε όλες τις περιπτώσεις χαμηλότερη απόδοση, επιβεβαιώνοντας ότι η επιλογή της αρχιτεκτονικής επηρεάζει σημαντικά την ικανότητα ανίχνευσης deepfake εικόνων. Η μελέτη αυτή καταδεικνύει τη σημασία της σωστής επιλογής συνόλου δεδομένων και μοντέλου, τονίζοντας πως η προσαρμογή της αρχιτεκτονικής στις ιδιαιτερότητες κάθε συνόλου δεδομένων μπορεί να βελτιώσει αισθητά την ακρίβεια της ταξινόμησης.

5. Μελλοντική Εργασία

Η έρευνα αυτή ενισχύει τη βιβλιογραφία γύρω από την ανίχνευση ψεύτικου πολυμεσικού υλικού, αναδεικνύοντας την αποτελεσματικότητα των συνελικτικών νευρωνικών δικτύων στη συγκεκριμένη εφαρμογή. Η παρούσα διπλωματική εργασία επικεντρώθηκε στην ανάπτυξη και αξιολόγηση ενός μοντέλου ανίχνευσης deepfake, καταδεικνύοντας ότι με σύγχρονες τεχνικές βαθιάς μάθησης είναι εφικτή η δημιουργία ενός αποδοτικού ταξινομητή. Το φαινόμενο των deepfakes αποτελεί μια αυξανόμενη απειλή για την αξιοπιστία της πληροφόρησης, ενώ οι προκλήσεις στην ανίχνευσή τους καθιστούν απαραίτητη την περαιτέρω έρευνα και βελτίωση των μεθόδων ανίχνευσης. Η παρούσα μελέτη ανέδειξε την αποτελεσματικότητα του EfficientNet στην κατηγοριοποίηση deepfake εικόνων και βίντεο και έδειξε πως η επιλογή των δεδομένων και η διαχείρισή τους παίζουν καθοριστικό ρόλο στη συνολική απόδοσή του ταξινομητή.

Παρότι τα αποτελέσματα ήταν ενθαρρυντικά, υπάρχουν αρκετές κατευθύνσεις που μπορούν να βελτιώσουν περαιτέρω την ανίχνευση ψεύτικου περιεχομένου. Πρωτίστως, η χρήση μιας πιο σύνθετης έκδοσης του EfficientNet, όπως το EfficientNet-B3 ή B4, θα μπορούσε να αποδώσει καλύτερα, αξιοποιώντας βαθύτερες αρχιτεκτονικές για τη μάθηση πιο σύνθετων χαρακτηριστικών. Ωστόσο, η μεγαλύτερη υπολογιστική απαίτηση αυτών των μοντέλων, αποτελεί έναν περιορισμό, καθώς απαιτούν περισσότερη μνήμη και μεγαλύτερο χρόνο εκπαίδευσης.

Μια δεύτερη πιθανή βελτίωση είναι η ανάπτυξη ενός υβριδικού συστήματος, όπου συνδυάζονται διαφορετικές αρχιτεκτονικές μοντέλων, ώστε να βελτιωθεί η ικανότητα του συστήματος να εντοπίζει περισσότερα μοτίβα ψεύτικης επεξεργασίας. Ο συνδυασμός διαφορετικών μεθόδων μπορεί να επιτρέψει πιο ισχυρή γενίκευση και μεγαλύτερη ανθεκτικότητα του ταξινομητή έναντι νέων, πιο εξελιγμένων deepfake τεχνικών.

Ένας ακόμη σημαντικός παράγοντας είναι η επέκταση και η καλύτερη διαχείριση των δεδομένων που χρησιμοποιούνται για την εκπαίδευση του συστήματος. Ένα ευρύτερο και πιο ποικιλόμορφο σύνολο δεδομένων, το οποίο περιλαμβάνει διαφορετικούς τύπους ψεύτικου περιεχομένου, θα μπορούσε να συμβάλει στη βελτίωση της ικανότητας γενίκευσης του συστήματος, καθιστώντας το πιο αποτελεσματικό σε πραγματικές συνθήκες.

Συνοψίζοντας, η αντιμετώπιση του φαινομένου DeepFake αποτελεί ένα πεδίο συνεχούς εξέλιξης, όπου οι τεχνικές παραγωγής ψεύτικου περιεχομένου βελτιώνονται διαρκώς. Η παρούσα εργασία απέδειξε ότι είναι εφικτή η αποτελεσματική ανίχνευση με διαθέσιμα εργαλεία και περιορισμένους πόρους, ωστόσο, η βελτίωση των μεθόδων και η εξέλιξη των τεχνικών ανίχνευσης αποτελεί απαραίτητο βήμα για την ενίσχυση της αξιοπιστίας της ψηφιακής πληροφόρησης.

Ακρωνύμια

ΤΝ	Τεχνητή Νοημοσύνη
ΣΝΔ	Συνελικτικά Νευρωνικά Δίκτυα
AI	Artificial Intelligence
AUC	Area Under Curve
CNN	Convolutional Neural Network
DF	DeepFake
DL	Deep Learning
FN	False Negative
FP	False Positive
GAN	Generative Adversarial Network
ML	Machine Learning
RNN	Recurrent Neural Network
TN	True Negative
TP	True Positive

Παράρτημα Α’

Κώδικας Μοντέλου

```
import os
import random
import numpy as np
import seaborn as sns
from sklearn.metrics import roc_auc_score, confusion_matrix, classification_report
from PIL import Image
import matplotlib.pyplot as plt
import torch
import torch.nn as nn
from torch.optim import Adam
from torchvision import transforms
from torch.utils.data import Dataset, DataLoader, random_split
import timm

# Set device (GPU if available, otherwise CPU)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(f"Using device: {device}")

# --------------- GLOBAL SEED FOR REPRODUCIBILITY ---------------
SEED = 1
torch.backends.cudnn.deterministic = True
torch.backends.cudnn.benchmark = False
random.seed(SEED)
torch.manual_seed(SEED)
np.random.seed(SEED)

def _init_fn(worker_id):
    worker_seed = SEED + worker_id
    np.random.seed(worker_seed)
    random.seed(worker_seed)
    torch.manual_seed(worker_seed)

if torch.cuda.is_available():
    torch.cuda.manual_seed(SEED)

# --------------- DATASET BUILDER ---------------
```

```
class Dataset_Builder(Dataset):
    def __init__(self, fake_dir, real_dir, transform=None):
        self.transform = transform
        self.fake_images = [os.path.join(fake_dir, img) for img in
os.listdir(fake_dir)]
        self.real_images = [os.path.join(real_dir, img) for img in
os.listdir(real_dir)]
        self.all_images = self.fake_images + self.real_images
        self.labels = [1] * len(self.fake_images) + [0] * len(self.real_images)  # 1
for fake, 0 for real

    def __len__(self):
        return len(self.all_images)

    def __getitem__(self, idx):
        image_path = self.all_images[idx]
        image = Image.open(image_path)
        if self.transform:
            image = self.transform(image)
        label = self.labels[idx]

        name_id = image_path.split('/')[-1].replace('.jpg', '')
        return image, label, name_id

# Function to split dataset into training and validation sets
def split_dataset(dataset, train_size, val_size, seed):
    torch.manual_seed(seed)
    train_data, val_data = random_split(dataset, [train_size, val_size])
    return train_data, val_data

# --------------- MODEL DEFINITION ---------------
class Network(nn.Module):
    def __init__(self, base_model_name="efficientnet_b0"):
        super(Network, self).__init__()

        base_model = timm.create_model(base_model_name, pretrained=True)
        self.features = base_model

        # Freeze initials layers
        for param in self.features.parameters():
            param.requires_grad = False

        # Adjust output layer for binary classification
        num_features = base_model.classifier.in_features
```

```
        base_model.classifier = nn.Linear(num_features, 1)

    def unfreeze_last_layers(self, num_layers=5):
        layers = list(self.features.children())
        for layer in layers[-num_layers:]:
            for param in layer.parameters():
                param.requires_grad = True

    def forward(self, x):
        x = self.features(x)
        return x

# --------------- DATA LOADING ---------------

# Set dataset paths
root_path_fake = '/kaggle/input/exp-data/EXP/FAKE'
root_path_real = '/kaggle/input/exp-data/EXP/REAL'

transform = transforms.Compose([
    transforms.Resize((450, 450)),  # input size
    transforms.ToTensor(),
])

# Load dataset
dataset = Dataset_Builder(root_path_fake, root_path_real, transform=transform)

data_size = len(dataset)
train_size = int(0.8 * data_size)
test_size = data_size - train_size
train_dataset, val_dataset = split_dataset(dataset, train_size, test_size, SEED)

# Create data loaders
train_loader = DataLoader(train_dataset, batch_size=32, shuffle=True, num_workers=0,
worker_init_fn=_init_fn)
val_loader = DataLoader(val_dataset, batch_size=32, shuffle=False, num_workers=0,
worker_init_fn=_init_fn)

# --------------- TRAINING CONFIGURATION ---------------
epochs = 10
lr = 0.0001

torch.manual_seed(SEED)
Net = Network(base_model_name="efficientnet_b0")
Net.unfreeze_last_layers(5)  # Unfreeze the last 5 layers
```

```
model = Net.to(device)
optimizer = Adam(model.parameters(), lr=lr, weight_decay=1e-3)
loss_fn = nn.BCEWithLogitsLoss()


# Lists for storing training metrics
train_losses = []
val_accuracies = []

# Early Stopping Parameters
patience = 3  # Number of epochs to wait for improvement
best_val_loss = float('inf')
trigger = 0

# --------------- TRAINING LOOP ---------------
for epoch in range(epochs):
    model.train()
    total_loss, corrects = 0, 0

    for batch_idx, (data, target, _) in enumerate(train_loader):
        data, targets = data.to(device), target.float().unsqueeze(1).to(device)
        optimizer.zero_grad()

        output = model(data)
        loss = loss_fn(output, targets)
        loss.backward()
        optimizer.step()
        total_loss += loss.item()
        preds = torch.round(torch.sigmoid(output))
        corrects += (preds == targets).sum().item()

    avg_loss = total_loss / len(train_loader.dataset)
    accuracy = 100. * corrects / len(train_loader.dataset)
    train_losses.append(avg_loss)

   # --------------- VALIDATON LOOP ---------------
    model.eval()
    val_loss, corrects = 0, 0
    all_preds = []
    all_probs = []
    all_targets = []
    with torch.no_grad():
        for data, target, _ in val_loader:
            data, targets = data.to(device), target.float().unsqueeze(1).to(device)
```

```
            output = model(data)
            loss = loss_fn(output, targets)
            val_loss += loss.item()
            preds = torch.round(torch.sigmoid(output))
            corrects += (preds == targets).sum().item()

            all_preds.extend(preds.cpu().numpy())
            all_probs.extend(torch.sigmoid(output).cpu().numpy())
            all_targets.extend(targets.cpu().numpy())

    avg_val_loss = val_loss / len(val_loader.dataset)
    val_accuracy = 100. * corrects / len(val_loader.dataset)
    val_accuracies.append(val_accuracy)

    # Print training progress
    print(f"Epoch {epoch + 1}/{epochs}")
    print("-" * 15)
    print(f"Train Loss: {avg_loss:.4f} train Acc: {accuracy:.4f}%")
    print(f"Validation Loss: {avg_val_loss:.4f} Validation Accuracy:
{val_accuracy:.2f}%")

 # Early Stopping
    if avg_val_loss < best_val_loss:
        best_val_loss = avg_val_loss
        trigger = 0
        torch.save(model.state_dict(), 'best_model.pth')  # Save best model
        print("Best model saved.")
    else:
        trigger += 1
        print(f"Early stopping trigger: {trigger}/{patience}")
        if trigger >= patience:
            print("Early stopping activated.")
            break

# ------------ Final Evaluation Metrics -----------------
auc = roc_auc_score(all_targets, all_probs)
tn, fp, fn, tp = confusion_matrix(all_targets, all_preds).ravel()
accuracy = (tp + tn) / (tp + tn + fp + fn)
sensitivity = tp / (tp + fn)
specificity = tn / (tn + fp)
precision = tp / (tp + fp)

print(f"\nFinal Metrics:")
```

```
print(f"Accuracy: {accuracy:.4f}")
print(f"AUC: {auc:.4f}")
print(f"Sensitivity (Recall): {sensitivity:.4f}")
print(f"Specificity: {specificity:.4f}")
print(f"Precision: {precision:.4f}")

# ------------ TRAINING LOSS & VALIDATION ACCURACY PLOT ---------------
plt.figure(figsize=(10, 6))
plt.plot(range(1, len(train_losses) + 1), train_losses, label="Training Loss",
marker="o", color="blue")
plt.xlabel("Epochs")
plt.ylabel("Loss", color="blue")
plt.tick_params(axis='y', labelcolor="blue")

ax2 = plt.gca().twinx()
ax2.plot(range(1, len(val_accuracies) + 1), val_accuracies, label="Validation
Accuracy", marker="o", linestyle="--", color="green")
ax2.set_ylabel("Accuracy (%)", color="green")
ax2.tick_params(axis='y', labelcolor="green")
plt.title("Training Loss and Validation Accuracy Over Epochs")
plt.grid()
plt.legend(loc="upper center")
plt.show()


# ------------ CONFUSION MATRIX HEATMAP ---------------
cm = confusion_matrix(all_targets, all_preds)

plt.figure(figsize=(6, 5))
sns.heatmap(cm, annot=True, fmt="d", cmap="Purples", xticklabels=["Real", "Fake"],
yticklabels=["Real", "Fake"])
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix")
plt.show()



# ------------ BATCH VISUALIZATION FUNCTION ---------------
def visualize_batch(images, labels, batch_size, title="Batch of Images"):
    plt.figure(figsize=(12, 12))
    for i in range(min(len(images), batch_size)):
        plt.subplot(3, 3, i + 1)
        img = images[i].cpu().numpy().transpose((1, 2, 0))
```

```
        plt.imshow(img)
        plt.title("Fake" if labels[i] == 1 else "Real")
        plt.axis("off")
    plt.suptitle(title)
    plt.show()

# Retrieve a batch of images
batch_size = 9
dataiter = iter(DataLoader(dataset, batch_size=batch_size, shuffle=True, num_workers=0,
worker_init_fn=_init_fn))
images, labels, _ = next(dataiter)

# Move images to CUDA device if available
if torch.cuda.is_available():
    images = images.cuda()

# Visualize the batch of images
visualize_batch(images, labels, batch_size)
```

Βιβλιογραφία

[1] L. Gaur, DeepFakes: Creation, Detection, and Impact, CRC Press, 2022.

[2] M. &. N. M. &. M. B. &. S. A. Rana, «Deepfake Detection: A Systematic Literature Review,» *IEEE Access,* 2022.

[3] S. Lyu, «Deepfake Detection: Current Challenges and Next Steps,» *EEE International Conference on Multimedia \& Expo Workshops (ICMEW),* 2020.

[4] M. M. E.-G. a. M. A. a. S. S. A. a. S. Sweidan, «A novel approach for detecting deep fake videos using graph neural network,» *Journal of Big Data,* τόμ. 11, pp. 1-27, 2024.

[5] M. Westerlund, «The Emergence of Deepfake Technology: A Review,» *Technology Innovation Management Review,* τόμ. 9, pp. 39-52, 2019.

[6] A. a. H. F. a. S. K. a. K. S. Boutadjine, «A Comprehensive Study on Multimedia DeepFakes,» *Conference: International Conference on Advances in Electronics, Control and Communication Systems,* 2023.

[7] A. S. Bahar Uddin Mahmud, «Deep Insights of Deepfake Technology : A Review,» *DUJASE ,* τόμ. 5, pp. 13-23, 2020.

[8] A. a. K. M. a. A. S. M. a. K. A. N. Malik, «DeepFake Detection for Human Face Images and Videos: A Survey,» *IEEE Access,* τόμ. 10, pp. 18757-18775, 2022.

[9] Y. W. a. B.-J. K. Bart van der Sloot, DEEPFAKES: THE LEGAL CHALLENGES OF A SYNTHETIC SOCIETY, Tilburg Institute for Law, Technology, and Society, 2021.

[10] S. Karnouskos, «Artificial Intelligence in Digital Media: The Era of Deepfakes,» *IEEE Transactions on Technology and Society,* τόμ. 1, pp. 138-147, 2020.

[11] D. A. a. C. R. a. F. F. a. G. C. Coccomini, «On the Generalization of Deep Learning Models in Video Deepfake Detection,» *Journal of Imaging,* τόμ. 9, 2023.

[12] D. S. A. a. N. S. S. a. z. A. a. M. M. a. M. Hazem, «{DeepFakeDG: A Deep Learning Approach for Deep Fake Detection and Generation,» *Journal of Computing and Communication,* 2023.

[13] R. M. S. P. M. P. Alexandros Haliassos, «Leveraging Real Talking Faces via Self-Supervision for Robust Forgery Detection,» 2022.

[14] Z. C. A. O. Chao Feng, «Self-Supervised Video Forensics by Audio-Visual Anomaly Detection,» 2023.

[15] D. a. G. M. a. G. M. Arora, «Diving deep in Deep Convolutional Neural Network,» σε *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, 2020, pp. 749-751.

[16] M. M. Taye, «Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions,» *Computation,* τόμ. 11, 2023.

[17] M. Krichen, «Convolutional Neural Networks: A Survey,» *Computers,* τόμ. 12, 2023.

[18] X. a. L. Z. a. Z. L. Pan, «Comprehensive Survey of State-of-the-Art Convolutional Neural Network Architectures and Their Applications in Image Classification,» *Innovations in Applied Engineering and Technology,* pp. 1-16, 2022.

[19] Z. A. a. J. Z. A. Haq, «Impact of Activation Functions and Number of Layers on the Classification of Fruits using CNN,» σε *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2021, pp. 227-231.

[20] Z. Z. Lei Zhao, «A improved pooling method for convolutional neural networks,» 2024.

[21] W. contributors, «Backpropagation,» Wikipedia, The Free Encyclopedia, 2025. [Ηλεκτρονικό].

Available: https://en.wikipedia.org/w/index.php?title=Backpropagation&oldid=1269494408.

[22] Z. a. L. F. a. Y. W. a. P. S. a. Z. J. Li, «A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects,» *IEEE Transactions on Neural Networks and Learning Systems,* τόμ. 33, pp. 6999-7019, 2022.

[23] H. W. a. X. Gu, «Towards dropout training for convolutional neural networks,» *Neural Networks,* τόμ. 71, pp. 1-10, 2015.

[24] L. Prechelt, «Early Stopping - But When?,» *Lecture Notes in Computer Science,* 2000.

[25] S. a. I. J. Mazilu, «L1 vs. L2 Regularization in Text Classification when Learning from Labeled Features,» σε *2011 10th International Conference on Machine Learning and Applications and Workshops*, 2011, pp. 166-171.

[26] J. B. B. P. J. L. R. H. M. W. C. C. F. Brian Dolhansky, «The DeepFake Detection Challenge (DFDC) Dataset,» *CoRR,* 2020.

[27] J. H. a. X. W. a. B. D. a. P. D. a. C. Xu, «DeepFake MNIST+: A DeepFake Facial Animation Dataset,» *CoRR,* 2021.

[28] CIPLAB, «Real and Fake Face Detection,» Kaggle, [Ηλεκτρονικό]. Available: https://www.kaggle.com/datasets/ciplab/real-and-fake-face-detection?select=real_and_fake_face.

[29] Q. V. L. Mingxing Tan, «EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,» *CoRR,* 2019.

[30] NVIDIA, «NVIDIA T4 Tensor Core GPU for AI Inference,» NVIDIA, [Ηλεκτρονικό]. Available: https://www.nvidia.com/en-eu/data-center/tesla-t4/.

[31] S. G. F. M. A. L. J. B. G. C. T. K. Z. L. N. G. L. A. A. D. A. K. E. Y. Z. D. M. R. A. T. S. C. Adam Paszke, «PyTorch: An Imperative Style, High-Performance Deep Learning Library,» *CoRR,*

2019.

[32] «NumPy,» [Ηλεκτρονικό]. Available: https://numpy.org/doc/stable/.

[33] «Matplotib,» [Ηλεκτρονικό]. Available: https://matplotlib.org/stable/index.html.

[34] X. Z. S. R. J. S. Kaiming He, «Deep Residual Learning for Image Recognition,» *CoRR,* 2015.