ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη Συστήματος Αναγνώρισης Παραποιημένων Εικόνων»

Των φοιτητών
Κωνσταντίνου Σιανίδη
Αρ. Μητρώου: 154535

Ελένης Πλήθου
Αρ. Μητρώου: 514118

Επιβλέπων
Κωτσάκης Ρήγας
Επίκουρος Καθηγητής

25 Μαΐου 2024

Τίτλος Δ.Ε.: Ανάπτυξη Συστήματος Αναγνώρισης Παραποιημένων Εικόνων
Κωδικός Δ.Ε.: 23332
Ονοματεπώνυμο φοιτητών: Κωνσταντίνος Σιανίδης, Ελένη Πλήθου
Ονοματεπώνυμο εισηγητή: Ρήγας Κωτσάκης
Ημερομηνία ανάληψης Δ.Ε.: 09-11-2023
Ημερομηνία περάτωσης Δ.Ε.: 25-05-2024

Βεβαιώνουμε ότι είμαστε οι συγγραφείς αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχαμε για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχουμε καταγράψει τις όποιες πηγές από τις οποίες κάναμε χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνουμε ότι αυτή η εργασία προετοιμάστηκε από εμάς προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

«Αφιερωμένη στους γονείς μας»

Πρόλογος

Η απόφαση να ασχοληθούμε με μια διπλωματική που θα επικεντρώνεται στην ανάπτυξη ενός μοντέλου νευρωνικού συνελικτικού δικτύου για την ανίχνευση εικόνων απομίμησης (deepfake) προήλθε από ένα βαθύ ενδιαφέρον για τη διασταύρωση της τεχνητής νοημοσύνης και της επεξεργασίας εικόνας. Η επικράτηση της τεχνολογίας deepfake στο σημερινό ψηφιακό κόσμο υπογραμμίζει την επείγουσα ανάγκη για ισχυρούς μηχανισμούς ανίχνευσης. Εμβαθύνοντας σε αυτόν τον τομέα, είχαμε ως στόχο όχι μόνο να εμβαθύνουμε στην κατανόηση των προηγμένων τεχνικών μηχανικής μάθησης αλλά και να συμβάλουμε στις συνεχιζόμενες προσπάθειες για την καταπολέμηση της παραπληροφόρησης και την διατήρηση της ακεραιότητας του ψηφιακού περιεχομένου. Μέσω αυτής της προσπάθειας, προσδοκούμε να αποκτήσουμε ανεκτίμητες γνώσεις σχετικά με την πολυπλοκότητα της ανίχνευσης deepfakes, να εξοπλιστούμε με πρακτικές δεξιότητες και να συνεισφέρουμε ουσιαστικά στον τομέα της μηχανικής μάθησης και της ασφάλειας στον κυβερνοχώρο.

Περίληψη

Η ανίχνευση παραποιημένων εικόνων που έχουν υποστεί επεξεργασία με την τεχνολογία deepfake είναι ζωτικής σημασίας λόγω του κίνδυνου παραπληροφόρησης. Τα deepfakes που απεικονίζουν πειστικά γεγονότα που δεν συνέβησαν ποτέ αποτελούν σοβαρές απειλές για διάφορους τομείς, συμπεριλαμβανομένης της πολιτικής, της δημοσιογραφίας και της ψυχαγωγίας. Αν δεν ελεγχθούν, μπορούν να διαβρώσουν την εμπιστοσύνη στα οπτικά μέσα και να διαστρεβλώσουν την πραγματικότητα, οδηγώντας σε κοινωνικά προβλήματα. Οι προκλήσεις στον εντοπισμό των deepfakes περιλαμβάνουν την ταχεία πρόοδο της τεχνολογίας τεχνητής νοημοσύνης, που καθιστά όλο και πιο δύσκολη τη διάκριση μεταξύ πραγματικού και ψεύτικου περιεχομένου. Ωστόσο, οι τεχνικές μηχανικής μάθησης, ιδίως τα Συνελικτικά Νευρωνικά Δίκτυα (CNNs), προσφέρουν πολλά υποσχόμενες λύσεις. Εκπαιδεύοντας τα σε μεγάλα σύνολα δεδομένων τόσο αυθεντικών όσο και παραποιημένων εικόνων, μπορούν να μάθουν να εντοπίζουν λεπτές ασυνέπειες ή τεχνουργήματα ενδεικτικά της χειραγώγησης deepfake. Έτσι, μέσω της προόδου στη μηχανική μάθηση και της ανάπτυξης ισχυρών αλγορίθμων ανίχνευσης, μπορούμε να μετριάσουμε τους κινδύνους που εγκυμονεί η τεχνολογία deepfake και να προστατεύσουμε την ακεραιότητα των οπτικών μέσων.

Στην παρούσα διπλωματική εργασία πραγματοποιούμε τον σχεδιασμό και την υλοποίηση ενός μοντέλου μηχανικής μάθησης το οποίο είναι ικανό να εντοπίσει εάν υπάρχει κάποια deepfake επεξεργασία σε μια εικόνα που του δίνεται. Για την υλοποίηση του μοντέλου έπρεπε να βρούμε ένα μεγάλο σύνολο δεδομένων που περιέχει, τόσο αυθεντικές, όσο και παραποιημένες με deepfake εικόνες. Στην συνέχεια, ορίζοντας ποιες εικόνες είναι αυθεντικές και ποιες τροποποιημένες και κάνοντας τις σωστές παραμετροποιήσεις και δοκιμές, εκπαιδεύσαμε ένα μοντέλο όπου είναι ικανό στο να εντοπίσει τα μοτίβα για να βρίσκει τις διαφορές μεταξύ των παραποιημένων και αυθεντικών εικόνων με αποτέλεσμα να μπορεί να τα ξεχωρίσει ακόμα και σε αθέατα δεδομένα. Αφού υλοποιηθεί το μοντέλο το χρησιμοποιούμε σε μια εφαρμογή με διεπαφή χρήστη ώστε να μπορεί να το αξιοποιήσει με ευκολία ένας χρήστης. Τέλος, παρουσιάσαμε κάποιες λεπτομέρειες για την απόδοση του εκπαιδευομένου μοντέλου.

«Development of a Manipulation Image Recognition System»

Konstantinos Sianidis

Eleni Plithou

Abstract

The detection of falsified images processed with deepfake technology is crucial due to the risk of misinformation. Deepfakes depicting convincing events that never happened pose serious threats to a variety of sectors, including politics, journalism, and entertainment. If left unchecked, they can erode trust in visual media and distort reality, leading to social problems. Challenges in detecting deepfakes include the rapid advancement of artificial intelligence technology, which makes it increasingly difficult to distinguish between real and fake content. However, machine learning techniques, particularly Convolutional Neural Networks, offer promising solutions. By training them on large datasets of both authentic and fake images, they can learn to identify subtle inconsistencies or artifacts indicative of deepfake manipulation. Thus, through advances in machine learning and the development of powerful detection algorithms, we can mitigate the risks posed by deepfake technology and protect the integrity of visual media.

In this thesis, we carry out the design and implementation of a machine learning model capable of detecting whether there is any deepfake processing in an image given to it. To implement the model, we had to find a large dataset containing both authentic and deepfake manipulated images. Then, by defining which images are authentic and which are altered, and making the right parameterizations and tests, we trained a model capable of identifying patterns to find the differences between the altered and authentic images so that it can distinguish them even in unseen data. Once the model is implemented, we use it in a user interface application so that a user can easily utilize it. Finally, we presented some details on the performance of the trained model.

Ευχαριστίες

Η διπλωματική εργασία υλοποιήθηκε από τον προπτυχιακό φοιτητή Κωνσταντίνο Σιανίδη και την προπτυχιακή φοιτήτρια Ελένη Πλήθου του Τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου Ελλάδος, κατά το ακαδημαϊκό έτος 2023-2024, υπό την επίβλεψη του καθηγητή Κωτσάκη Ρήγα.

Θα θέλαμε από κοινού να εκφράσουμε τη βαθύτατη ευγνωμοσύνη μας στον καθηγητή μας, Κωτσάκη Ρήγα, που δέχτηκε να επιβλέψει την διπλωματική μας εργασία. Η αφοσίωση, η τεχνογνωσία και η αδιάλειπτη υποστήριξή του ήταν ανεκτίμητη καθ' όλη τη διάρκεια αυτού του ταξιδιού. Είμαστε ευγνώμονες για τον χρόνο που αφιέρωσε για να μας καθοδηγήσει και να προσφέρει ανεκτίμητες γνώσεις. Η ενθάρρυνση και η προθυμία του να βοηθήσει έχουν εμπλουτίσει την ακαδημαϊκή μας εμπειρία.

Είμαστε επίσης υπόχρεοι στους άλλους καθηγητές μας, τα μαθήματα των οποίων παρείχαν τις θεμελιώδεις γνώσεις που ήταν απαραίτητες για την υλοποίηση αυτής της διπλωματικής. Το πάθος τους για τη διδασκαλία και η δέσμευσή τους στην αριστεία έπαιξαν καθοριστικό ρόλο στη διαμόρφωση της ακαδημαϊκής μας εξέλιξης.

Επιπλέον, εκφράζουμε την ειλικρινή εκτίμησή μας στις οικογένειές μας για την αμέριστη υποστήριξή τους καθ’ όλη την διάρκεια των σπουδών μας, οικονομικά αλλά και συναισθηματικά. Η ενθάρρυνσή τους τόσο στις χαρούμενες όσο και στις δύσκολες στιγμές υπήρξε πηγή δύναμής και είμαστε βαθύτατα ευγνώμονες για την πίστη τους σε εμάς.

Τέλος, θα θέλαμε να ευχαριστήσουμε όλους όσους συνέβαλαν με οποιαδήποτε ιδιότητα στην ολοκλήρωση της παρούσας διπλωματικής. Η υποστήριξη και η ενθάρρυνσή σας υπήρξαν καθοριστικές σε αυτή την προσπάθεια.

Περιεχόμενα

Κατάλογος εικόνων

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
RGB	Red Green Blue
CNN	Convolutional Neural Network
LUV	Lightness, Chromaticity coordinate, representing color information on the u* axis.
	Chromaticity coordinate, representing color information on the v* axis.
CM	Chromaticity Coordinates
CCV	Color Coherence Vector
HSV	Hue, Saturation, and Value
HMMD	Hue, Max, Min, and Diff
GAN	Generative Adversarial Networks
RNN	Recurrent Neural Networks
OS	Operating System
IMG	Image
EX	Exception
GUI	Graphical user interface
UI	User Interface
H5	Hierarchical Data Format version 5
QSS	Qt Style Sheets
JPG	Joint Photographic Experts Group
PNG	Portable Network Graphics
GB	Gigabyte
JSON	JavaScript Object Notation

Κεφάλαιο 1ο: Η εργασία

1.1 Εισαγωγή

Στο εισαγωγικό αυτό το κεφάλαιο θα παρουσιαστούν οι λόγοι που οδήγησαν στην ιδέα για την δημιουργία της παρούσας διπλωματικής εργασίας. Επιπλέον, θα περιγραφεί η δομή της εργασίας καθώς και οι στόχοι που επιδιώκει να επιτύχει.

1.2 Περιγραφή ιδέας

Η ιδέα για την ανάπτυξη ενός μοντέλου για τον εντοπισμό εικόνων απομίμησης (deepfakes) προκλήθηκε από την ταχεία εξάπλωση της τεχνολογίας απομίμησης και τις επιπτώσεις της στην κοινωνία. Καθώς παρατηρούσαμε τις εξελίξεις στην τεχνητή νοημοσύνη, συνειδητοποιήσαμε τις δυνατότητες τόσο για ευεργετικές όσο και για κακόβουλες χρήσεις αυτών των τεχνολογιών. Η τεχνολογία Deepfake, η οποία επιτρέπει τη δημιουργία εξαιρετικά ρεαλιστικών συνθετικών εικόνων και βίντεο, ξεχώρισε ως σημαντική ανησυχία.

Η στιγμή που προκάλεσε το ενδιαφέρον μας ήταν η παρακολούθηση αρκετών περιπτώσεων όπου τα deepfakes χρησιμοποιήθηκαν για τη διάδοση παραπληροφόρησης, τη χειραγώγηση της κοινής γνώμης και την παραβίαση της ιδιωτικής ζωής των ατόμων. Τα περιστατικά αυτά ανέδειξαν την επιτακτική ανάγκη για αποτελεσματικά εργαλεία για τον εντοπισμό και τον μετριασμό των κινδύνων που συνδέονται με το περιεχόμενο deepfake. Η αυξανόμενη πολυπλοκότητα αυτών των αλγορίθμων, οι οποίοι μπορούν να παράγουν περιεχόμενο δυσδιάκριτο από το αυθεντικό, μας έκανε να καταλάβουμε πόσο επείγει η επίλυσή αυτού του ζητήματος.

Το υπόβαθρό μας στον τομέα των υπολογιστών και τη μηχανική μάθηση μας παρείχε την απαραίτητη τεχνική βάση για την αντιμετώπιση αυτής της πρόκλησης. Αναγνωρίσαμε ότι ένα μοντέλο, με την ικανότητά να μαθαίνει αυτόματα και να εξάγει χαρακτηριστικά από εικόνες, θα μπορούσε να αποτελέσει ένα ισχυρό εργαλείο για την ανίχνευση λεπτών τεχνουργημάτων και ασυνεπειών που υπάρχουν σε εικόνες με πλαστογραφία. Παρακινηθήκαμε από τη δυνατότητα να συμβάλλουμε σε ένα ασφαλέστερο ψηφιακό περιβάλλον, όπου η αυθεντικότητα των οπτικών μέσων θα μπορεί να επαληθεύεται αξιόπιστα.

Η ιδέα της δημιουργίας ενός μοντέλου για την ανίχνευση deepfake καθοδηγήθηκε επίσης από τις ευρύτερες επιπτώσεις σε διάφορους τομείς, όπως η δημοσιογραφία, η επιβολή του νόμου και η προσωπική ασφάλεια. Με την ανάπτυξη ενός ισχυρού συστήματος ανίχνευσης, είχαμε ως στόχο να παράξουμε ένα εργαλείο που θα μπορούσε να βοηθήσει στη διατήρηση της ακεραιότητας του ψηφιακού περιεχομένου και στην προστασία ατόμων και οργανισμών από τις επιζήμιες επιπτώσεις της τεχνολογίας deepfake.

1.3 Οργάνωση ενοτήτων

Στο πρώτο κεφάλαιο γίνεται μια εισαγωγή στην παρούσα διπλωματική εργασία. Παρουσιάζονται οι λόγοι οι οποίοι μας οδήγησαν στην πραγματοποίηση της, οργανώνονται οι ενότητες ώστε να έχουμε μια πρώτη εικόνα για την δομή της και τέλος παρουσιάζονται οι στόχοι που θέλουμε να πετύχουμε.

Στο δεύτερο κεφάλαιο, μιλάμε για το πρόβλημα της παραποίησης της εικόνας. Δίνεται αρχικά ο ορισμός της εικόνας και εξετάζεται το πρόβλημα της παραποίησης της καθώς και οι λόγοι για τους οποίους συμβαίνει.

Στο τρίτο κεφάλαιο μαθαίνουμε κάποιες πληροφορίες για την εικόνα και τα χαρακτηριστικά της, πράγμα βασικό για την κατανόηση του προβλήματος αλλά και της λύσης του.

Στο τέταρτο κεφάλαιο παρουσιάζουμε το πρόβλημα αλλά και μερικές τεχνολογίες που το βοηθάει να εξαπλωθεί ώστε να κατανοήσουμε την σημαντικότητά του.

Στο πέμπτο κεφάλαιο μιλάμε για την επίλυση του προβλήματος. Αναφέρουμε και σχετικές προσπάθειες που έχουν γίνει στο παρελθόν αλλά και κάποιες πληροφορίες για την δική μας προσπάθεια.

Στο έκτο κεφάλαιο θα γίνει αναλυτική παρουσίαση της υλοποίησης του μοντέλου μας καθώς και η επεξήγηση διαφόρων εννοιών και πληροφοριών που πρέπει να γνωρίζουμε σχετικά με αυτό.

Στο έβδομο κεφάλαιο θα γίνει παρουσίαση της εφαρμογής μέσα από την οποία χρησιμοποιούμε το μοντέλο μας, τόσο σε οπτικό όσο και σε λειτουργικό επίπεδο.

Στο όγδοο κεφάλαιο θα γίνει αναφορά στο σύνολο των δεδομένων όπου χρησιμοποιήσαμε για την δημιουργία του μοντέλου μας καθώς και κάποιες προκλήσεις που είχαμε σχετικά με αυτό.

Στο ένατο κεφάλαιο θα παρουσιαστεί η αποτελεσματικότητα του μοντέλου κάνοντας μια αξιολόγηση στην απόδοσή του.

Τέλος, στο δέκατο κεφάλαιο θα γίνει μια αναφορά στα συμπεράσματα που βγάλαμε καθώς και σε μελλοντικές προτάσεις και προκλήσεις που υπάρχουν.

1.4 Στόχοι

Ο πρωταρχικός στόχος αυτής της διπλωματικής εργασίας είναι η ανάπτυξη ενός αποτελεσματικού μοντέλου, ικανού να ανιχνεύει με ακρίβεια εικόνες απομίμησης. Αυτό περιλαμβάνει τον σχεδιασμό ενός μοντέλου που όχι μόνο επιτυγχάνει υψηλή ακρίβεια και αξιοπιστία στη διάκριση μεταξύ γνήσιων και παραποιημένων εικόνων, αλλά και γενικεύεται καλά σε διάφορες τεχνικές και σύνολα δεδομένων deepfake. Επιπλέον, η εργασία στοχεύει στη βελτιστοποίηση του μοντέλου για αποδοτικότητα, εξισορροπώντας τις υπολογιστικές απαιτήσεις με την ταχύτητα ανίχνευσης, ώστε να είναι κατάλληλο για εφαρμογές στον πραγματικό κόσμο. Ένας άλλος σημαντικός στόχος είναι να συνεισφέρει πολύτιμες γνώσεις και μεθοδολογίες στον τομέα της ανίχνευσης deepfake, παρέχοντας μια βάση για περαιτέρω έρευνα και ανάπτυξη. Τελικά, η διπλωματική εργασία φιλοδοξεί να δημιουργήσει ένα πρακτικό εργαλείο που μπορεί να χρησιμοποιηθεί για την επαλήθευση της αυθεντικότητας του οπτικού περιεχομένου και την προστασία από τις επιζήμιες επιπτώσεις των deepfakes, διατηρώντας έτσι την ακεραιότητα και την αξιοπιστία των ψηφιακών μέσων.

1.5 Επίλογος

Το παρόν κεφάλαιο παρέχει μια ολοκληρωμένη επισκόπηση της ιδέας της διπλωματικής εργασίας, παρουσιάστηκε η δομή της οργανωμένη σε κεφάλαια και τέλος προσδιορίστηκε ο στόχος εκπόνησής της.

Κεφάλαιο 2ο: Το πρόβλημα της παραποίησης της εικόνας

2.1 Εισαγωγή

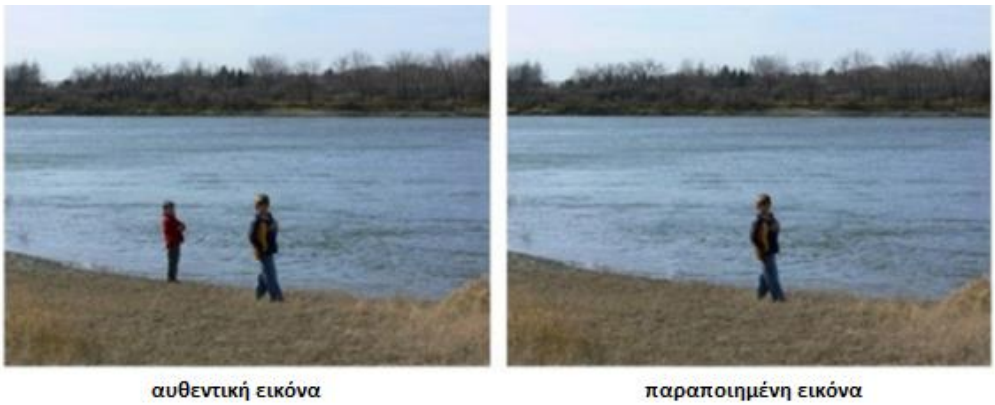
Σε αυτό το κεφάλαιο δίνεται ο ορισμός της εικόνας, εξετάζεται συνοπτικά το πρόβλημα της παραποίησης της, καθώς και οι λόγοι για τους οποίους συμβαίνει. Τέλος, αναφέρονται οι μελλοντικές προοπτικές των εφαρμογών αναγνώρισης παραποίησης εικόνας.

2.2 Ορισμός της εικόνας

Η εικόνα είναι μια οπτική αναπαράσταση ενός αντικειμένου, μιας σκηνής ή μιας έννοιας, η οποία αποτυπώνεται ή δημιουργείται με διάφορα μέσα, όπως η φωτογραφία, η ζωγραφική, το σχέδιο ή η ψηφιακή απόδοση. Αποτελείται από έναν πίνακα εικονοστοιχείων σε ψηφιακή μορφή, κάθε εικονοστοιχείο αντιπροσωπεύει ένα συγκεκριμένο χρώμα και μια συγκεκριμένη τιμή έντασης, τα οποία μαζί σχηματίζουν μια συνεκτική εικόνα αντιληπτή από το ανθρώπινο μάτι. Οι εικόνες μπορεί να είναι δισδιάστατες, όπως οι φωτογραφίες και οι οθόνες, ή τρισδιάστατες, όπως τα ολογράμματα και ορισμένα είδη τέχνης, και χρησιμοποιούνται για ένα ευρύ φάσμα σκοπών, όπως η επικοινωνία, η τέχνη, η τεκμηρίωση και η επιστημονική ανάλυση [1].

2.3 Ορισμός της παραποιημένης εικόνας

Μια παραποιημένη εικόνα αναφέρεται σε μια εικόνα που έχει τροποποιηθεί ή αλλοιωθεί με κάποιο τρόπο ώστε να δημιουργηθεί μια ψευδής αναπαράσταση της πραγματικότητας. Αυτό μπορεί να γίνει μέσω διαφόρων τρόπων όπως ένα λογισμικό επεξεργασίας φωτογραφιών, αλγόριθμοι βαθιάς μάθησης ή άλλες μορφές χειραγώγησης εικόνας. Οι παραποιημένες εικόνες μπορούν να δημιουργηθούν για διάφορους σκοπούς, όπως η διάδοση παραπληροφόρησης, η δημιουργία προπαγάνδας ή ακόμη και για ψυχαγωγία. Οι αλγόριθμοι ανίχνευσης πλαστών εικόνων έχουν σχεδιαστεί για να αναλύουν εικόνες και να εντοπίζουν σημάδια χειραγώγησης, όπως ασυνέπειες στον φωτισμό, την προοπτική ή άλλα οπτικά στοιχεία [2].



Εικόνα 2.1: Παραποίηση εικόνας [3]

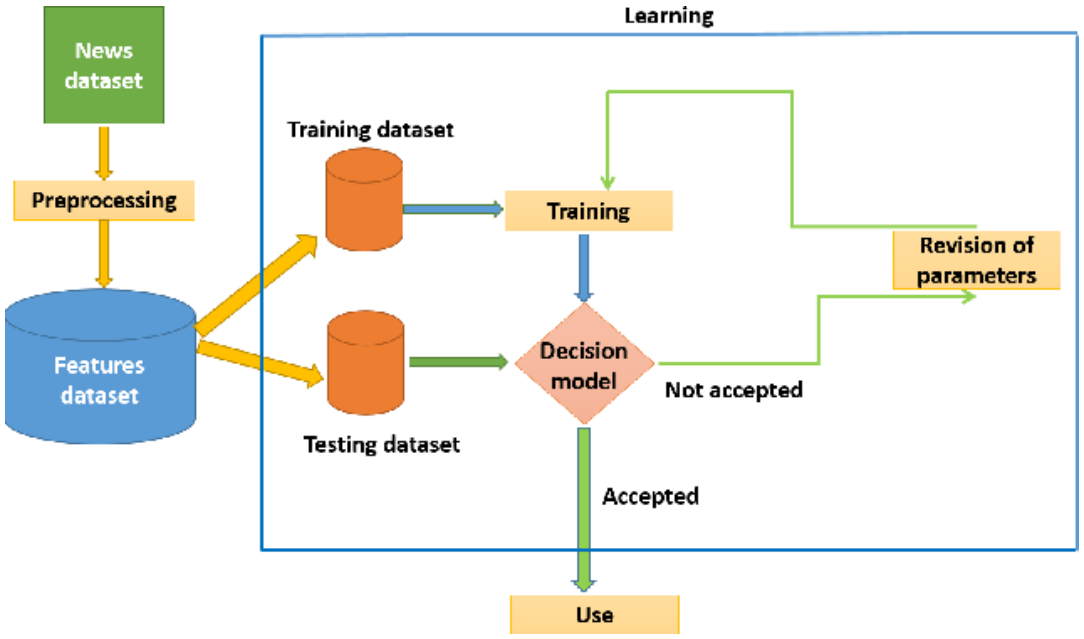
2.4 Εξέταση του προβλήματος

Η επικράτηση των προηγμένων εργαλείων επεξεργασίας εικόνας στο σημερινό ψηφιακό τοπίο έχει δημιουργήσει μια σειρά νέων προκλήσεων που σχετίζονται με την παραπληροφόρηση. Με την εμφάνιση των εξελιγμένων τεχνικών deepfakes που βασίζονται στην τεχνητή νοημοσύνη, παράλληλα με τα λιγότερο εξελιγμένα αλλά παραπλανητικά cheapfakes, τα άτομα είναι όλο και περισσότερο σε θέση να χειραγωγούν τα μέσα ενημέρωσης για διάφορους κακόβουλους σκοπούς. Οι δραστηριότητες αυτές κυμαίνονται από τη διάδοση ψευδών πληροφοριών και την κατασκευή επιβλαβούς περιεχομένου, όπως το πορνό εκδίκησης, έως τη διάπραξη απάτης [4] [5].

Τα deepfakes αποτελούν ένα χαρακτηριστικό παράδειγμα της πιθανής κατάχρησής αυτών των εργαλείων. Τα πρόσωπα, ειδικότερα, έχουν τεράστια σημασία σε αυτές τις παραποιήσεις λόγω της κεντρικής τους θέσης στην ανθρώπινη επικοινωνία και των προηγμένων τεχνικών που είναι διαθέσιμες για την ανακατασκευή και τον εντοπισμό τους [6].

Οι μέθοδοι χειραγώγησης προσώπων συνήθως χωρίζονται σε δύο κύριες κατηγορίες. Την χειραγώγηση της έκφρασης του προσώπου και την χειραγώγηση της ταυτότητας του προσώπου. Ένα παράδειγμα τέτοιων τεχνικών είναι το Face2Face που αναπτύχθηκε από τους Thies et al., όπου επιτρέπει τη μεταφορά εκφράσεων προσώπου σε πραγματικό χρόνο από ένα άτομο σε ένα άλλο. Αυτή η δυνατότητα όχι μόνο εγείρει ανησυχίες σχετικά με την ιδιωτικότητα και τη συναίνεση, αλλά υπογραμμίζει και την ανάγκη για ισχυρούς μηχανισμούς κατά της κατάχρησης αυτής της τεχνολογίας [6].

Επιπλέον, τεχνολογίες που έχουν την δυνατότητα να δημιουργούν κινούμενες εκφράσεις προσώπου με βάση ηχητικές εισόδους, θολώνουν ακόμα περισσότερο τα όρια μεταξύ πραγματικότητας και απάτης. Καθώς αυτές οι τεχνικές γίνονται όλο και πιο εξελιγμένες και προσβάσιμες, καθίσταται επιτακτική ανάγκη να αναπτυχθούν μηχανισμοί για τον εντοπισμό και την αντιμετώπιση των αρνητικών τους επιπτώσεων, συμπεριλαμβανομένης της πιθανής δυσφήμισης, της παραπληροφόρησης και της διάβρωσης της εμπιστοσύνης στο ψηφιακό περιεχόμενο [6].



Εικόνα 2.2: Αρχιτεκτονική συστήματος ανακάλυψης παραποιημένων εικόνων [2].

2.5 Επίλυση του προβλήματος

Είναι ζωτικής σημασίας να αναπτυχθούν ισχυρά συστήματα ικανά να ανιχνεύουν και να επισημαίνουν τα παραποιημένα μέσα ενημέρωσης που θα μπορούσαν δυνητικά να προκαλέσουν βλάβη. Η ταχεία επέκταση και η ευρεία χρήση των πλατφορμών κοινωνικών μέσων μεγαλώνουν την ανάγκη για τον εντοπισμό λύσεων, καθώς οι εν λόγω πλατφόρμες δίνουν όλο και μεγαλύτερη έμφαση στο οπτικό περιεχόμενο [7]. Ως εκ τούτου, η επένδυση σε τεχνολογίες και μεθοδολογίες για τον εντοπισμό και την καταπολέμηση της διάδοσης των παραπλανητικών μέσων είναι επιτακτική ανάγκη για τη διασφάλιση της ακεραιότητας του διαδικτυακού λόγου και τον μετριασμό των αρνητικών συνεπειών της παραπληροφόρησης [4].

Στην προσπάθεια μας για την επίλυση του προβλήματος δεν πρέπει να αγνοούνται οι προκλήσεις και οι περιορισμοί των σημερινών μεθόδων ανίχνευσης ψηφιακής πλαστογραφίας. Μια πρόκληση αποτελεί η πρόθεση. Πολλοί χειρισμοί εικόνων είναι καλοήθεις και γίνονται για αθώους σκοπούς, όπως το ρετουσάρισμα φωτογραφιών ενός ταξιδιού ή η αφαίρεση των κόκκινων ματιών. Η διάκριση μεταξύ των αβλαβών αλλοιώσεων και των επιβλαβών χειρισμών, οι οποίοι αλλάζουν την ερμηνεία της εικόνας, είναι ζωτικής σημασίας. Οι τρέχουσες μέθοδοι ανίχνευσης πλαστογραφίας συχνά αποτυγχάνουν να διακρίνουν μεταξύ αυτών των σκοπών. Μια ακόμα πρόκληση αφορά την ανθεκτικότητα. Ενώ οι υπάρχοντες αλγόριθμοι ανίχνευσης πλαστογραφίας υπερέχουν στον εντοπισμό γνωστών τεχνικών χειραγώγησης αναλύοντας στατιστικά στοιχεία σε επίπεδο εικονοστοιχείου και συμπίεσης, μπορεί να δυσκολεύονται με νέου είδους τεχνολογίες. Αυτό μπορεί να συμβαίνει όταν προσαρμόζονται υπερβολικά σε γνωστά μοτίβα χειραγώγησης και να αποτυγχάνουν να ανιχνεύσουν εξελιγμένες πλαστογραφίες που αναπτύσσονται με νέες τεχνικές. Ως εκ τούτου, υπάρχει ανάγκη για πιο ισχυρές μεθόδους ανίχνευσης ικανές να εντοπίζουν παραποιήσεις πέραν των γνωστών μοτίβων [4].

2.6 Μελλοντικές προοπτικές

Οι μελλοντικές προοπτικές των εφαρμογών που βασίζονται σε αλγόριθμους βαθιάς μάθησης είναι πολύ μεγάλες. Σκοπός είναι η ανάπτυξη υβριδικών αλγορίθμων που ενσωματώνουν ειδικά στρώματα για να ενισχύσουν τον εντοπισμό αλλοιωμένων μέσων ώστε να γίνουν ιδιαίτερα αποδοτικοί για χρήση σε πλατφόρμες μέσων κοινωνικής δικτύωσης, κυβερνητικές υπηρεσίες και στο ευρύ κοινό. Τέτοιες εφαρμογές θα διαδραματίσουν καθοριστικό ρόλο στην επαλήθευση της αυθεντικότητας των μέσων ενημέρωσης και στην παρακολούθηση του εικονικού χώρου για τυχόν χειραγώγηση. Αυτές οι εξελίξεις αποσκοπούν στην προώθηση ενός ασφαλέστερου ψηφιακού περιβάλλοντος και υπόσχονται εφαρμογές σε διάφορους τομείς, συμπεριλαμβανομένων των πλατφόρμων κοινωνικής δικτύωσης και των κυβερνητικών υπηρεσιών [8].

2.7 Επίλογος

Το απόσπασμα συζητά τόσο για τις εικόνες γενικότερα αλλά για τις παραποιημένες εικόνες ειδικότερα. Συνεχίζει με τις προκλήσεις που θέτει η διάδοση των παραποιημένων εικόνων, ιδίως με την άνοδο τεχνολογιών όπως το deepfake. Περιγράφει τη σημασία της ανίχνευσης και αντιμετώπισης των πλαστών εικόνων, ιδίως στο πλαίσιο της παραπληροφόρησης και της πιθανής βλάβης. Συζητούνται οι προκλήσεις που υπάρχουν στις τρέχουσες μεθόδους ανίχνευσης όπως η διάκριση μεταξύ καλοηθών και επιβλαβών

χειρισμών ή οι δυσκολίες με τις νέες, εξελιγμένες τεχνικές παραποίησης. Τέλος, συζητούνται οι μελλοντικές προοπτικές με στόχο ένα ασφαλέστερο ψηφιακό περιβάλλον.

Κεφάλαιο 3ο: Χαρακτηριστικά εικόνας

3.1 Εισαγωγή

Προκειμένου να μπορέσει ένας υπολογιστής να κατανοήσει σημασιολογικά μια εικόνα, πρέπει πρώτα να εξάγει και να μοντελοποιήσει αποτελεσματικά οπτικά χαρακτηριστικά και όχι να βασίζεται στην ανθρώπινη γνώση. Η εξαγωγή και η επιλογή των σωστών οπτικών χαρακτηριστικών χαμηλού επιπέδου όπως το χρώμα, η υφή και το σχήμα, είναι ζωτικής σημασίας στις εργασίες επεξεργασίας εικόνας. Τα περισσότερα συστήματα χρησιμοποιούν αυτά τα χαρακτηριστικά και η απόδοσή τους εξαρτάται σε μεγάλο βαθμό από την ποιότητα των χαρακτηριστικών που εξάγονται [9].

3.2 Χαρακτηριστικά χρώματος

Το χρώμα είναι ένα από τα βασικότερα χαρακτηριστικά των εικόνων, που ορίζεται μέσα σε έναν συγκεκριμένο χρωματικό χώρο ή μοντέλο. Τα χρωματικά χαρακτηριστικά των εικόνων αναλύονται συχνά με τη χρήση διαφόρων χρωματικών χώρων, όπως οι RGB, HSV, LUV και HSL, HMMD [5] καθένας από τους οποίους προσφέρει μοναδικές αναπαραστάσεις των χρωμάτων. Μόλις επιλεγεί ένας χρωματικός χώρος, μπορούν να εξαχθούν χρωματικά χαρακτηριστικά από εικόνες ή περιοχές τους. Τεχνικές όπως τα ιστογράμματα χρώματος [7] παρέχουν πληροφορίες σχετικά με την κατανομή των χρωμάτων σε μια εικόνα, ενώ περιγραφείς όπως το διάνυσμα συνοχής χρώματος (CCV) [10] καταγράφουν τη χωρική συνοχή των χρωμάτων. Άλλα σημαντικά χρωματικά χαρακτηριστικά περιλαμβάνουν τις χρωματικές ροπές (CM) που είναι ένα από τα απλούστερα αλλά ιδιαίτερα αποτελεσματικά χαρακτηριστικά [11], και τα χρωματικά συσχετογραμμάτα [12]. Επιπλέον, οι χρωματικές συντεταγμένες προσφέρουν τυποποιημένες αναπαραστάσεις των χρωμάτων με βάση τα χρωματικά χαρακτηριστικά τους. Αυτές οι μέθοδοι επιτρέπουν συλλογικά την ολοκληρωμένη ανάλυση και τον χειρισμό της χρωματικής πληροφορίας σε εικόνες για ένα ευρύ φάσμα εφαρμογών.

Εικόνα 3.1: Χρωματικός χώρος RGB VS HSV [13]

3.3 Χαρακτηριστικά υφής

Η υφή της εικόνας στις ψηφιακές εικόνες αναφέρεται στα οπτικά μοτίβα και τις διαφοροποιήσεις στην ένταση ή το χρώμα που δίνουν σε μια επιφάνεια την χαρακτηριστική της εμφάνιση και αίσθηση. Η υφή μπορεί να περιγράψει τις λεπτές λεπτομέρειες, την ποιότητα της επιφάνειας και τη χωρική διάταξη των εικονοστοιχείων σε μια εικόνα, συμβάλλοντας στην αντίληψη των ιδιοτήτων και της δομής των απεικονιζόμενων αντικειμένων.

Στην ψηφιακή επεξεργασία εικόνας, η υφή συχνά αναλύεται με διάφορες μαθηματικές και στατιστικές μεθόδους για την ποσοτικοποίηση αυτών των μοτίβων και παραλλαγών. Τεχνικές όπως οι πίνακες συνεμφάνισης σε επίπεδο γκρι, τα τοπικά δυαδικά μοτίβα και οι μετασχηματισμοί κυματιδίων χρησιμοποιούνται συνήθως για τη σύλληψη και την περιγραφή χαρακτηριστικών υφής. Αυτά τα χαρακτηριστικά μπορεί να περιλαμβάνουν χαρακτηριστικά, όπως η ομαλότητα, η τραχύτητα, η κανονικότητα, και η αντίθεση.

Η ανάλυση υφής είναι ζωτικής σημασίας σε πολλές εφαρμογές, όπως η κατάτμηση εικόνων και η αναγνώριση αντικειμένων. Για παράδειγμα, στην ιατρική απεικόνιση, η υφή μπορεί να βοηθήσει στη διάκριση μεταξύ υγιών και ασθενών ιστών, ενώ στην τηλεπισκόπηση μπορεί να βοηθήσει στην ταξινόμηση των τύπων κάλυψης γης. Στην όραση υπολογιστών, η κατανόηση και η ανάλυση της υφής είναι απαραίτητη για εργασίες όπως η αναγνώριση προσώπου, αλλά και στον εντοπισμό μοτίβων σε εικόνες. Οι βασικότερες μέθοδοι για την εξαγωγή χαρακτηριστικών υφής είναι αυτές που εξάγουν χωρικά χαρακτηριστικά και αυτές που εξάγουν χαρακτηριστικά του φάσματος [14].

Εικόνα 3.2: Υφές εικόνων [15]

3.3.1 Μέθοδοι εξαγωγής χωρικών χαρακτηριστικών υφής

Οι μέθοδοι που εξάγουν χωρικά χαρακτηριστικά υφής εστιάζουν στην ανάλυση των μοτίβων και των μεταβολών στις τιμές έντασης των εικονοστοιχείων απευθείας στο χωρικό πεδίο της εικόνας. Αυτές οι μέθοδοι αποσκοπούν στην καταγραφή της τοπικής δομής και των επαναλαμβανόμενων μοτίβων που υπάρχουν σε διάφορες περιοχές της εικόνας. Εξετάζοντας τον τρόπο με τον οποίο οι τιμές των εικονοστοιχείων μεταβάλλονται στο χώρο, οι τεχνικές αυτές μπορούν να προσδιορίσουν σημαντικά χαρακτηριστικά υφής, όπως η ομαλότητα, η τραχύτητα, η κανονικότητα και η κατευθυντικότητα. Η χωρική ανάλυση υφής βασίζεται στην παραδοχή ότι οι υφές συχνά ορίζονται από τις χωρικές σχέσεις μεταξύ των εντάσεων των εικονοστοιχείων. Τα χαρακτηριστικά που εξάγονται μπορούν να περιλαμβάνουν στατιστικά μέτρα των κατανομών της έντασης των εικονοστοιχείων, καθώς και ιδιότητες που περιγράφουν τη γεωμετρική διάταξη των τιμών των εικονοστοιχείων. Αυτές οι μέθοδοι

είναι ιδιαίτερα χρήσιμες για εργασίες που απαιτούν την κατανόηση των τοπικών ιδιοτήτων υφής, όπως η κατάτμηση εικόνων και η αναγνώριση αντικειμένων [1].

3.3.2 Μέθοδοι εξαγωγής φασματικών χαρακτηριστικών υφής

Οι μέθοδοι που εξάγουν φασματικά χαρακτηριστικά υφής λειτουργούν στο πεδίο της συχνότητας και όχι στο πεδίο του χώρου. Αυτές οι μέθοδοι περιλαμβάνουν μετασχηματισμό της εικόνας στο πεδίο της συχνότητας για την ανάλυση της περιοδικότητας και των συνιστωσών συχνότητας της υφής. Με την εξέταση του φασματικού περιεχομένου της εικόνας, οι τεχνικές αυτές μπορούν να εντοπίσουν επαναλαμβανόμενα μοτίβα και τον προσανατολισμό των υφών. Η ανάλυση στο πεδίο της συχνότητας βοηθά στην κατανόηση των υποκείμενων χαρακτηριστικών συχνότητας των υφών, τα οποία συχνά σχετίζονται με την κλίμακα και την κατεύθυνση των μοτίβων υφής. Η φασματική ανάλυση υφής καταγράφει τις παγκόσμιες ιδιότητες υφής με την αποσύνθεση της εικόνας στις συνιστώσες συχνότητας που την απαρτίζουν, παρέχοντας μια διαφορετική προοπτική σε σύγκριση με τις χωρικές μεθόδους. Αυτή η προσέγγιση είναι χρήσιμη για τον εντοπισμό υφών που παρουσιάζουν κανονικά, επαναλαμβανόμενα μοτίβα και για εφαρμογές που απαιτούν ανάλυση πολλαπλών κλιμάκων, όπως η ταξινόμηση υφών, η αναγνώριση μοτίβων και η συμπίεση εικόνων [1].

3.4 Χαρακτηριστικά σχήματος

Το σχήμα χρησιμεύει ως κρίσιμη ένδειξη για τον άνθρωπο κατά τον εντοπισμό και την αναγνώριση αντικειμένων του πραγματικού κόσμου, κωδικοποιώντας απλές γεωμετρικές μορφές, όπως ευθείες γραμμές προς διάφορες κατευθύνσεις. Οι τεχνικές για την εξαγωγή χαρακτηριστικών σχήματος μπορούν να κατηγοριοποιηθούν, σε γενικές γραμμές, σε δύο ομάδες [16], τις μεθόδους που βασίζονται στο περίγραμμα και τις μεθόδους που βασίζονται στην περιοχή. Τόσο οι μέθοδοι με βάση το περίγραμμα όσο και οι μέθοδοι με βάση την περιοχή έχουν τα πλεονεκτήματά τους και επιλέγονται με βάση τις ειδικές απαιτήσεις της εφαρμογής. Σε κάποιες περιπτώσεις ο συνδυασμός αυτών των προσεγγίσεων μπορεί να οδηγήσει σε πιο ισχυρά συστήματα ανάλυσης και αναγνώρισης σχήματος.

Επιπλέον, οι χωρικές σχέσεις διαδραματίζουν σημαντικό ρόλο στην επεξεργασία εικόνας, παρέχοντας πληροφορίες σχετικά με τις θέσεις των αντικειμένων εντός μιας εικόνας ή τις σχέσεις μεταξύ των αντικειμένων. Η εξέταση αυτή περιλαμβάνει συνήθως δύο σενάρια: την απόλυτη χωρική θέση των περιοχών [17] και τις σχετικές θέσεις των περιοχών [18] [19].

Εικόνα 3.3: Μέθοδος βασισμένη στο περίγραμμα [20].

3.4.1 Μέθοδοι βασισμένες στο περίγραμμα

Οι μέθοδοι που βασίζονται στο περίγραμμα εστιάζουν στην εξαγωγή και ανάλυση του ορίου ή του περιγράμματος ενός σχήματος. Αυτές οι μέθοδοι εξετάζουν κυρίως την περίμετρο ενός αντικειμένου, καταγράφοντας τις γεωμετρικές του ιδιότητες και παραλλαγές. Η θεμελιώδης ιδέα πίσω από τις μεθόδους που βασίζονται στο περίγραμμα είναι ότι το όριο ενός σχήματος περιέχει σημαντικές πληροφορίες σχετικά με τη μορφή και τη δομή του. Αναλύοντας το περίγραμμα, μπορεί κανείς να αντλήσει διάφορα χαρακτηριστικά, όπως η καμπυλότητα, ο προσανατολισμός και η συνολική γεωμετρία του σχήματος. Αυτές οι μέθοδοι είναι ιδιαίτερα αποτελεσματικές για τη διάκριση σχημάτων με βάση τα εξωτερικά χαρακτηριστικά τους και χρησιμοποιούνται συνήθως σε εφαρμογές όπως η αναγνώριση αντικειμένων, η ταξινόμηση και η κατάτμηση εικόνων. Η ανάλυση μπορεί να περιλαμβάνει τον προσδιορισμό της ομαλότητας, της συνέχειας και της πολυπλοκότητας του ορίου, τα οποία είναι απαραίτητα για την κατανόηση της οπτικής εμφάνισης του σχήματος και τον εντοπισμό μοναδικών μοτίβων. Οι μέθοδοι που βασίζονται στο περίγραμμα είναι ιδιαίτερα χρήσιμες για την καταγραφή λεπτομερών πληροφοριών ορίων και χρησιμοποιούνται συχνά στην αναγνώριση αντικειμένων και στην αντιστοίχιση σχημάτων [1].

3.4.2 Μέθοδοι με βάση την περιοχή

Οι μέθοδοι που βασίζονται στην περιοχή εστιάζουν στην ανάλυση ολόκληρης της περιοχής που καταλαμβάνει ένα σχήμα, λαμβάνοντας υπόψη την κατανομή και τη διάταξη όλων των εικονοστοιχείων εντός των ορίων του σχήματος. Αυτές οι μέθοδοι αποσκοπούν στην καταγραφή των εσωτερικών χαρακτηριστικών ενός σχήματος, παρέχοντας μια ολοκληρωμένη εικόνα της γεωμετρίας και της δομής του. Ο πρωταρχικός στόχος των μεθόδων που βασίζονται στην περιοχή είναι η κατανόηση του σχήματος εξετάζοντας τις χωρικές σχέσεις και την κατανομή των εικονοστοιχείων και όχι μόνο το περίγραμμα. Αυτό περιλαμβάνει τον υπολογισμό διαφόρων στατιστικών μέτρων και περιγραφικών δεικτών που αντικατοπτρίζουν τη συνολική μορφή, το μέγεθος, τον προσανατολισμό και άλλες ιδιότητες του σχήματος. Οι μέθοδοι με βάση την περιοχή είναι χρήσιμες σε εφαρμογές όπου οι εσωτερικές ιδιότητες του σχήματος είναι ζωτικής σημασίας, όπως η αναγνώριση προτύπων, η κατάτμηση εικόνων και η ταξινόμηση. Εστιάζοντας σε ολόκληρη την περιοχή, αυτές οι μέθοδοι μπορούν να παρέχουν πληροφορίες σχετικά με τη συμμετρία του σχήματος, τη συμπαγή μορφή και άλλες ιδιότητες που είναι απαραίτητες για τη διάκριση μεταξύ διαφορετικών σχημάτων και την κατανόηση των οπτικών χαρακτηριστικών τους. Οι μέθοδοι που βασίζονται στην περιοχή παρέχουν μια πιο ολιστική άποψη του σχήματος και είναι χρήσιμες σε εφαρμογές όπου η συνολική γεωμετρία και η κατανομή των εικονοστοιχείων είναι σημαντικές, όπως στην αναγνώριση προτύπων και την κατάτμηση εικόνων [1].

3.5 Επίλογος

Αυτό το κεφάλαιο διερευνά τα θεμελιώδη χαρακτηριστικά των εικόνων, εστιάζοντας στα χαρακτηριστικά του χρώματος, της υφής και του σχήματος. Όπως είδαμε, τα χαρακτηριστικά χρώματος, που ορίζονται σε χρωματικούς χώρους όπως οι RGB, LUV, HSV και HMMD, περιλαμβάνουν ιστογράμματα χρώματος, χρωματικές ροπές, διανύσματα χρωματικής συνοχής και διαγράμματα χρωματικής συσχέτισης. Τα χαρακτηριστικά υφής εξάγονται από ομάδες εικονοστοιχείων χρησιμοποιώντας χωρικές ή φασματικές μεθόδους και τα χαρακτηριστικά σχήματος με μεθόδους που βασίζονται στο περίγραμμα ή στην περιοχή. Επιπλέον, οι χωρικές σχέσεις παρέχουν κρίσιμες

πληροφορίες σχετικά με τις θέσεις και τις αλληλεπιδράσεις των αντικειμένων σε μια εικόνα, περιλαμβάνοντας τόσο απόλυτες όσο και σχετικές χωρικές θέσεις.

Κεφάλαιο 4ο: Παρουσίαση του προβλήματος

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο παρουσιάζονται διάφορες μέθοδοι παραποίησης εικόνας και αναλύονται οι πιο συχνά χρησιμοποιούμενοι τρόποι για την αντιμετώπιση του συγκεκριμένου προβλήματος.

4.2 Ορισμός του deepfake

Το deepfake αναφέρεται σε έναν τύπο συνθετικών μέσων, όπου η τεχνητή νοημοσύνη, και συγκεκριμένα οι τεχνικές βαθιάς μάθησης, χρησιμοποιούνται για τη δημιουργία εξαιρετικά ρεαλιστικών αλλά ψεύτικων εικόνων, ήχου ή βίντεο. Ο όρος συνδυάζει τις λέξεις "deep learning" και "fake", αντανακλώντας την προέλευσή του από εξελιγμένους αλγορίθμους μηχανικής μάθησης. Στο GitHub (πλατφόρμα που φιλοξενεί έργα ανάπτυξης λογισμικού), υπάρχουν πολυάριθμες δημόσιες υλοποιήσεις για την δημιουργία deepfakes, όπως το FakeApp και το Faceswap.

Τα deepfakes δημιουργούνται μέσω μιας διαδικασίας που περιλαμβάνει την εκπαίδευση μοντέλων βαθιάς μάθησης σε εκτεταμένα σύνολα δεδομένων εικόνων ή βίντεο ενός προσώπου. Αυτή η εκπαίδευση χρησιμοποιεί συνήθως Generative Adversarial Networks (GANs), τα οποία αποτελούνται από δύο νευρωνικά δίκτυα: τη γεννήτρια και τον διαχωριστή. Η γεννήτρια δημιουργεί συνθετικά μέσα, προσπαθώντας να αναπαραγάγει την ομοιότητα του προσώπου-στόχου, ενώ ο διαχωριστής αξιολογεί την αυθεντικότητα των παραγόμενων μέσων. Μέσω επαναληπτικής ανατροφοδότησης, η γεννήτρια βελτιώνει τα αποτελέσματά της έως ότου τα ψεύτικα μέσα ενημέρωσης γίνουν σχεδόν δυσδιάκριτα από το πραγματικό υλικό. Η διαδικασία περιλαμβάνει διάφορα στάδια, συμπεριλαμβανομένης της συλλογής δεδομένων, της εκπαίδευσης του μοντέλου και της τελικής σύνθεσης, όπου τα παραγόμενα μέσα αναμειγνύονται απρόσκοπτα με το αρχικό υλικό. Αυτό περιλαμβάνει εξελιγμένες τεχνικές για την αντιστοίχιση των εκφράσεων του προσώπου, του φωτισμού και άλλων στοιχείων του περιβάλλοντος, ώστε να διασφαλιστεί ότι το τελικό προϊόν είναι εξαιρετικά ρεαλιστικό. Τα deepfakes μπορούν να χρησιμοποιηθούν για διάφορους σκοπούς, από νόμιμες εφαρμογές στην ψυχαγωγία έως πιο κακόβουλες χρήσεις όπως η παραπληροφόρηση και η κλοπή ταυτότητας [21].

Εικόνα 4.1: Δημιουργία deepfake [22]

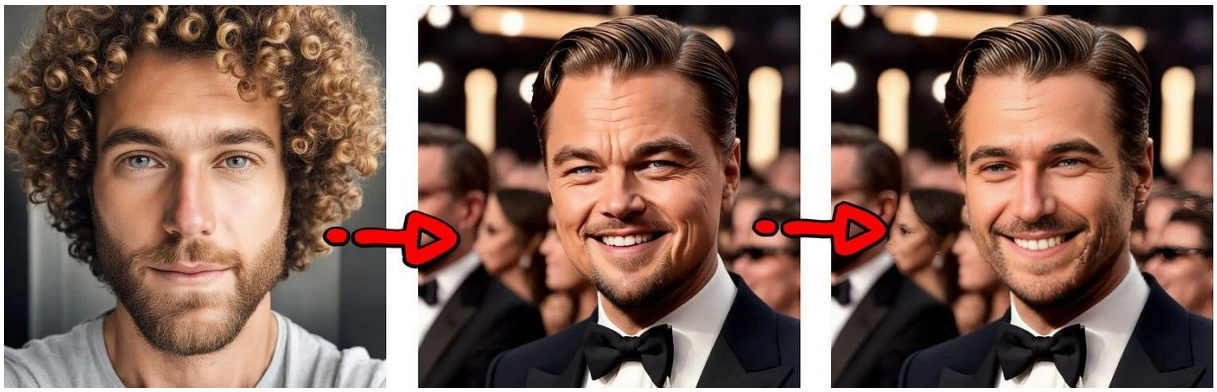
4.2.1 Face2Face

Το Face2Face λειτουργεί καταγράφοντας τις κινήσεις του προσώπου από μια πηγή και μεταφέρνοντάς τες σε πραγματικό χρόνο σε έναν στόχο. Αρχικά, ανιχνεύονται σημεία αναφοράς του προσώπου τόσο στην πηγή όσο και στο στόχο. Στη συνέχεια, ένας αλγόριθμος μεταφοράς εκφράσεων προσώπου υπολογίζει τις διαφορές στις κινήσεις του προσώπου μεταξύ των αντίστοιχων σημείων αναφοράς στην πηγή και στο στόχο. Αυτές οι διαφορές χρησιμοποιούνται για τη σύνθεση μιας νέας εικόνας προσώπου για το στόχο, που μιμείται τις εκφράσεις του προσώπου της πηγής. Τέλος, η συνθετική εικόνα προσώπου αναμειγνύεται απρόσκοπτα στο πρόσωπο-στόχο, με αποτέλεσμα μια ρεαλιστική απεικόνιση του προσώπου-στόχου με εκφράσεις από το πρόσωπο-πηγή. Αυτή η διαδικασία επιτρέπει τον χειρισμό των κινήσεων και των εκφράσεων, επιτρέποντας την πειστική αναπαράσταση του προσώπου και τη μεταφορά εκφράσεων [6].

Εικόνα 4.2: Παράδειγμα Face2Face [23]

4.2.2 Face Swap

Το Face Swap είναι μια μέθοδος βασισμένη στα γραφικά που χρησιμοποιείται για την μεταφορά της περιοχής προσώπου από μια προέλευση σε έναν προορισμό. Λειτουργεί με την εξαγωγή της περιοχής του προσώπου με βάση τα αραιά εντοπισμένα χαρακτηριστικά προσώπου. Αυτά τα χαρακτηριστικά καθοδηγούν την προσαρμογή ενός τρισδιάστατου πρότυπου μοντέλου με τη χρήση blendshapes (τρόπος παραμόρφωσης της γεωμετρίας). Στην συνέχεια, το μοντέλο προβάλλεται στην εικόνα προορισμού ελαχιστοποιώντας την διαφορά μεταξύ του προβαλλόμενου σχήματος και των εντοπισμένων χαρακτηριστικών, χρησιμοποιώντας τις υφές της εικόνας προέλευσης. Τέλος, το μοντέλο που αποδίδεται αναμιγνύεται απρόσκοπτα με την εικόνα και εφαρμόζεται διόρθωση χρώματος. Τα βήματα αυτά επαναλαμβάνονται για όλα τα ζεύγη καρέ πηγής και στόχου μέχρι να ολοκληρωθεί το βίντεο. Αυτή η υλοποίηση είναι υπολογιστικά ελαφριά και μπορεί να εκτελεστεί αποτελεσματικά στη CPU [6].



Εικόνα 4.3: Παράδειγμα face swap [24]

4.2.3 Face morphing

Η μορφοποίηση προσώπου λειτουργεί αρχικά με τον εντοπισμό αντίστοιχων σημείων αναφοράς ή σημείων-κλειδιών του προσώπου σε δύο ή περισσότερες εικόνες, συνήθως χρησιμοποιώντας τεχνικές όπως η ανίχνευση χαρακτηριστικών προσώπου ή η χειροκίνητη ανίχνευση. Στη συνέχεια, υπολογίζεται

ο γεωμετρικός μετασχηματισμός μεταξύ αυτών των σημείων, συχνά χρησιμοποιώντας τεχνικές όπως η τριγωνοποίηση Delaunay. Με την ομαλή παρεμβολή των χαρακτηριστικών του προσώπου και των χρωμάτων μεταξύ των εικόνων πηγής και στόχου σε μια σειρά καρέ, επιτυγχάνεται μια απρόσκοπτη μετάβαση, με αποτέλεσμα ένα μορφοποιημένο πρόσωπο που συνδυάζει χαρακτηριστικά και από τις δύο αρχικές εικόνες. Αυτή η διαδικασία μπορεί να βελτιωθεί περαιτέρω με τεχνικές για να διασφαλιστεί η ομαλή κίνηση και ο ρεαλισμός στο μορφοποιημένο αποτέλεσμα [25].

Εικόνα 4.4: Παράδειγμα face morphing [26]

4.3 Το πρόβλημα της ανίχνευσης

Παρά την πρόοδο στην ανίχνευση πλαστών εικόνων με χρήση μηχανικής μάθησης, εξακολουθούν να υπάρχουν αρκετές προκλήσεις και περιορισμοί. Μία από τις κύριες προκλήσεις είναι η ανάπτυξη ισχυρών μοντέλων που μπορούν να ανιχνεύσουν εξελιγμένες παραποιήσεις εικόνων. Μια άλλη πρόκληση είναι η ανάγκη για μεγάλα σύνολα δεδομένων τόσο πραγματικών όσο και πλαστών εικόνων για την αποτελεσματική εκπαίδευση των μοντέλων μηχανικής μάθησης. Επιπλέον, οι ηθικές επιπτώσεις της χρήσης τέτοιων μοντέλων, ιδίως σε τομείς όπως η πολιτική και τα μέσα μαζικής ενημέρωσης, πρέπει επίσης να εξεταστούν προσεκτικά.

Η ανίχνευση πλαστών εικόνων με χρήση μηχανικής μάθησης αποτελεί ένα δύσκολο και ταχέως εξελισσόμενο ερευνητικό πεδίο. Η υπάρχουσα βιβλιογραφία αναδεικνύει τις δυνατότητες διαφόρων τεχνικών μηχανικής μάθησης, συμπεριλαμβανομένων των CNNs, GANs και RNNs, στην ανίχνευση πλαστών εικόνων. Ωστόσο, απαιτείται περαιτέρω έρευνα για την ανάπτυξη πιο ισχυρών και αποτελεσματικών μοντέλων και για την αντιμετώπιση των ηθικών επιπτώσεων της χρήσης τους.

4.4 Επίλογος

Παρά την πρόοδο στη μηχανική μάθηση για την ανίχνευση πλαστών εικόνων, εξακολουθούν να υπάρχουν σημαντικές προκλήσεις. Η ανάπτυξη ισχυρών μοντέλων για τον εντοπισμό εξελιγμένων πλαστογραφιών και η ανάγκη για εκτεταμένα σύνολα δεδομένων πραγματικών και πλαστών εικόνων αποτελούν πρωταρχικά εμπόδια. Οι ηθικές επιπτώσεις, ιδίως στην πολιτική και τα μέσα ενημέρωσης, περιπλέκουν περαιτέρω το ζήτημα. Το παρόν κεφάλαιο εξετάζει κάποιες τεχνικές χειραγώγησης εικόνων, όπως τα deepfakes, Face2Face και Face Swap. Όπως είδαμε, τα deepfakes περιλαμβάνουν την αντικατάσταση προσώπου με χρήση βαθιάς μάθησης, με δημοφιλή εργαλεία όπως το FakeApp και το

Faceswap. Το Face2Face μεταφέρει εκφράσεις από μια προέλευση σε έναν προορισμού, ενώ το Face Swap χρησιμοποιεί μεθόδους που βασίζονται σε γραφικά για τη μεταφορά περιοχών προσώπου. Συζητούνται οι προκλήσεις και οι εξελίξεις στον εντοπισμό πλαστών εικόνων, τονίζοντας την ανάγκη για συνεχή βελτίωση των μεθοδολογιών ανίχνευσης για την καταπολέμηση των εξελιγμένων τεχνικών παραγωγής πλαστών εικόνων.

Κεφάλαιο 5ο: Επίλυση του προβλήματος

5.1 Εισαγωγή

Στο παρόν κεφάλαιο αναλύονται πιθανές λύσεις στο πρόβλημα της ανίχνευσης παραποιημένων εικόνων. Δίνεται μια σύντομη αναφορά σε προηγούμενες εργασίες οι οποίες επιχείρησαν να λύσουν το πρόβλημα. Τέλος, δίνεται συνοπτικά η μεθοδολογία της παρούσας εργασίας.

5.2 Ανίχνευση παραποιημένων γενικών εικόνων

Η ανίχνευση παραποιημένων γενικών εικόνων περιλαμβάνει την διαδικασία εξακρίβωσης του κατά πόσο μία εικόνα έχει παραποιηθεί ή αλλοιωθεί με οποιονδήποτε τρόπο ώστε να παράγει μία παραπλανητική ή ανακριβή απεικόνιση της πραγματικότητας. Αυτή η μορφή ανίχνευσης βρίσκει ευρεία εφαρμογή σε τομείς όπως η εγκληματολογία, η δημοσιογραφία και η εποπτεία των μέσων κοινωνικής δικτύωσης, με στόχο τον εντοπισμό εικόνων που έχουν παραποιηθεί ή αλλοιωθεί για κακόβουλες προθέσεις, όπως η διάδοση ψευδών ειδήσεων, προπαγάνδας ή παραπληροφόρησης. Οι τεχνικές για την ανίχνευση παραποιημένων γενικών εικόνων περιλαμβάνουν διάφορες προσεγγίσεις, όπως η διεξοδική εξέταση των ασυμφωνιών στο φωτισμό και στις σκιές, ο εντοπισμός παρατυπιών στα μοτίβα εικονοστοιχείων και η σύγκριση της εικόνας με καθιερωμένες αυθεντικές εικόνες ή εικόνες αναφοράς. Ορισμένοι αλγόριθμοι χρησιμοποιούν μεθοδολογίες μηχανικής μάθησης για την ανάλυση εκτεταμένων συνόλων δεδομένων που περιλαμβάνουν τόσο γνήσιες όσο και παραποιημένες εικόνες, ενισχύοντας έτσι την ακρίβεια της ανίχνευσης. Παρόλα αυτά είναι σημαντικό να αναγνωρίσουμε ότι καμία μεμονωμένη μέθοδος ή αλγόριθμος δεν μπορεί να επιτύχει απόλυτη ακρίβεια στην ανίχνευση όλων των μορφών παραποιημένων εικόνων, ενώ με την τεχνολογική πρόοδο κλιμακώνεται και η πολυπλοκότητα των τεχνικών για την δημιουργία πειστικών παραποιημένων εικόνων. Κατά συνέπεια, είναι επιτακτική ανάγκη να χρησιμοποιηθεί ένας συνδυασμός τεχνικών και ανθρώπινης εμπειρογνωμοσύνης για τον αποτελεσματικό εντοπισμό παραποιημένων εικόνων και τον περιορισμό της διάδοσης τους [27].

5.3 Ανίχνευση παραποιημένων εικόνων χωρίς χρήση μηχανικής μάθησης

Το ζήτημα παραποίησης εικόνας λαμβάνει όλα και μεγαλύτερες διαστάσεις σε αρκετούς τομείς, όπως παράδειγμα στα μέσα κοινωνικής δικτύωσης. Υπάρχουν, ωστόσο, διάφορες τεχνικές για την ανίχνευση πλαστών εικόνων.

Μια τεχνική είναι η αντίστροφη αναζήτηση εικόνων. Με τη χρήση μηχανών αντίστροφης αναζήτησης εικόνων, όπως το Google Images ή το TinEye, γίνεται έλεγχος αν μια εικόνα εμφανίζεται σε άλλους ιστότοπους ή πλατφόρμες μέσων κοινωνικής δικτύωσης. Εάν η εικόνα βρίσκεται σε πολλά μέρη, αυτό μπορεί να υποδεικνύει ότι δεν είναι πρωτότυπη ή ότι έχει παραποιηθεί.

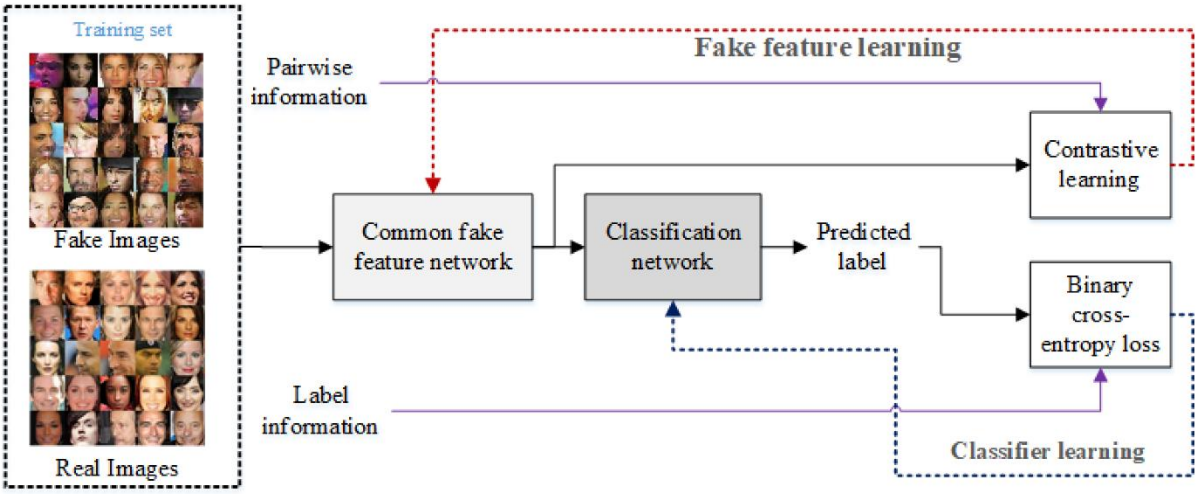
Μια αποτελεσματική τεχνική είναι και ο έλεγχος των μεταδεδομένων: Οι ψηφιακές εικόνες περιέχουν μεταδεδομένα με πληροφορίες σχετικά με το πότε και πού τραβήχτηκε η φωτογραφία, καθώς και τον τύπο της φωτογραφικής μηχανής ή της συσκευής που χρησιμοποιήθηκε. Η ανάλυση αυτών των

μεταδεδομένων μπορεί να αποκαλύψει ασυμφωνίες ή ασυνέπειες που υποδηλώνουν ότι η εικόνα έχει αλλοιωθεί.

Επίσης μπορεί να πραγματοποιηθεί και μια ανάλυση του περιεχομένου της εικόνας. Υπάρχουν, εργαλεία λογισμικού τα οποία μπορούν να αναλύσουν το περιεχόμενο μιας εικόνας για να προσδιορίσουν αν έχει γίνει αλλοίωση. Για παράδειγμα, αυτά τα εργαλεία μπορούν να εντοπίσουν ασυνέπειες στο φωτισμό ή το χρώμα που υποδεικνύουν ψηφιακές αλλοιώσεις.

Μια ακόμα τεχνική είναι η επαλήθευση πηγής. Η επαλήθευση της αυθεντικότητας μιας εικόνας συχνά περιλαμβάνει τον εντοπισμό της αρχικής πηγής. Αυτό μπορεί να γίνει επικοινωνώντας με το άτομο που δημοσίευσε την εικόνα ή χρησιμοποιώντας διαδικτυακά εργαλεία για την επαλήθευση της πηγής της.

Τέλος χρήσιμη μπορεί να είναι και η γνώμη κάποιου εμπειρογνώμονα. Η συμβουλή εμπειρογνωμόνων σε τομείς όπως η εγκληματολογία ή η ανάλυση εικόνας μπορεί να είναι απαραίτητη για τον προσδιορισμό της γνησιότητας μιας εικόνας. Αυτοί οι επαγγελματίες διαθέτουν εξειδικευμένες γνώσεις και εργαλεία για να εντοπίζουν σημάδια χειραγώγησης ή άλλα ζητήματα [2].



Εικόνα 5.1: Ανίχνευση παραποιημένων εικόνων [2]

5.4 Μηχανική μάθηση

Η μηχανική μάθηση είναι ένας τομέας της τεχνητής νοημοσύνης, που επιτρέπει στα συστήματα να μαθαίνουν από μοτίβα δεδομένων και να λαμβάνουν αποφάσεις ή προβλέψεις χωρίς ρητό προγραμματισμό. Στον πυρήνα της, οι αλγόριθμοι μηχανικής μάθησης επαναλαμβάνουν δεδομένα, προσαρμόζοντας τις παραμέτρους τους για να βελτιστοποιήσουν την απόδοση σε μια δεδομένη εργασία. Αυτή η επαναληπτική διαδικασία επιτρέπει στις μηχανές να διακρίνουν πολύπλοκα μοτίβα και σχέσεις μέσα στα δεδομένα, διευκολύνοντας εφαρμογές που κυμαίνονται από συστήματα συστάσεων έως την επεξεργασία φυσικής γλώσσας. Με την έλευση της βαθιάς μάθησης, ενός υποσυνόλου της μηχανικής μάθησης που χρησιμοποιεί νευρωνικά δίκτυα με πολλαπλά επίπεδα, οι μηχανές έχουν επιτύχει αξιοσημείωτα επιτεύγματα στην αναγνώριση ομιλίας, την αυτόνομη οδήγηση και την ιατρική διάγνωση, μεταξύ άλλων. Καθώς οι αλγόριθμοι γίνονται πιο εξελιγμένοι και τα σύνολα δεδομένων μεγαλώνουν, οι δυνατότητες της μηχανικής μάθησης να φέρει επανάσταση σε διάφορους κλάδους

συνεχίζουν να επεκτείνονται, υποσχόμενες καινοτομίες που επαναπροσδιορίζουν τις αλληλεπιδράσεις ανθρώπου-μηχανής και εξορθολογίζουν τις διαδικασίες [28].

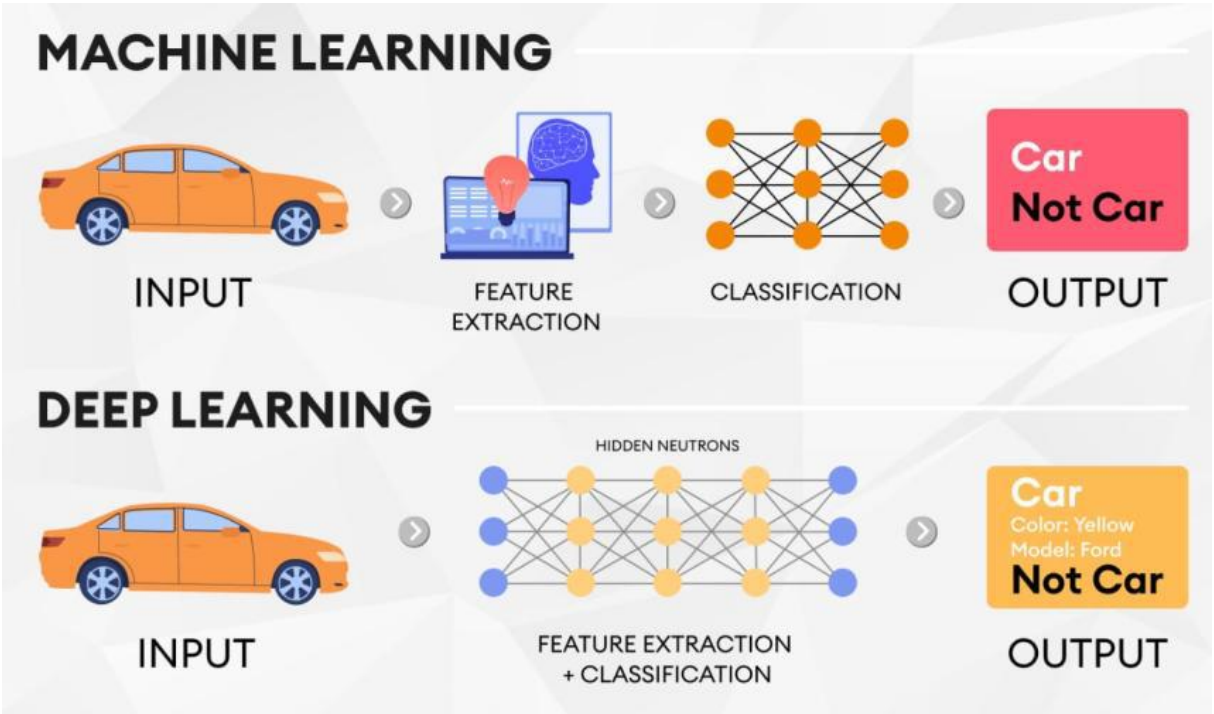
5.5 Ανίχνευση παραποιημένων εικόνων με χρήση μηχανικής μάθησης

Η αξιοποίηση της μηχανικής μάθησης για την αναγνώριση εικόνας περιλαμβάνει την καθοδήγηση ενός αλγόριθμου να διακρίνει αντικείμενα ή μοτίβα μέσα σε εικόνες. Η διαδικασία αυτή αφορά ένα σύνολο μεθοδολογιών, όπως η μάθηση με επίβλεψη, η μάθηση χωρίς επίβλεψη και η βαθιά μάθηση.

Η μάθηση με επίβλεψη συνεπάγεται την εκπαίδευση του αλγορίθμου σε ένα σύνολο δεδομένων με ετικέτες, όπου κάθε εικόνα είναι επισημασμένη με μία συγκεκριμένη ετικέτα που υποδεικνύει το απεικονιζόμενο αντικείμενο ή μοτίβο. Μέσω αυτής της διαδικασίας, ο αλγόριθμος αφομοιώνει τις σχέσεις μεταξύ των μοτίβων εικόνας και των αντίστοιχων ετικετών τους, μαθαίνοντας έτσι να αναγνωρίζει με ακρίβεια τα αντικείμενα.

Αντίθετα η μάθηση χωρίς επίβλεψη λειτουργεί σε μη επισημασμένα σύνολα δεδομένων, καθιστώντας την ικανή στην αποκάλυψη λανθάνουσας μορφής μοτίβων και συσχετίσεων σε εκτεταμένα σύνολα δεδομένων.

Η βαθιά μάθηση, ένα υποσύνολο της μηχανικής μάθησης, χρησιμοποιεί νευρωνικά δίκτυα για την εξαγωγή χαρακτηριστικών από εικόνες. Τα συνελικτικά νευρωνικά δίκτυα (CNN) είναι ιδιαίτερα αποτελεσματικά, αξιοποιώντας πολλαπλά στρώματα επεξεργασίας για την κατανόηση περίπλοκων λεπτομερειών όπως αιχμές, γωνίες και υφές σε εικόνες. Μόλις ο αλγόριθμος εκπαιδευτεί, μπορεί να αναγνωρίζει με επάρκεια αντικείμενα σε νέες εικόνες αναλύοντας τα χαρακτηριστικά τους και αντιστοιχίζοντας τα με αναγνωρισμένα μοτίβα.



Εικόνα 5.2: Αναγνώριση εικόνας [2]

5.6 Σχετικές εργασίες

Η ανίχνευση παραποιημένων εικόνων με χρήση μηχανικής μάθησης αποτελεί μια δύσκολη και αναδυόμενη ερευνητική περιοχή λόγω της αυξανόμενης διαθεσιμότητας και ευκολίας χρήσης των εργαλείων επεξεργασίας εικόνων. Τα τελευταία χρόνια, έχουν διεξαχθεί πολυάριθμες μελέτες για την ανάπτυξη αυτοματοποιημένων συστημάτων που μπορούν να ανιχνεύουν πλαστές εικόνες χρησιμοποιώντας τεχνικές μηχανικής μάθησης. Η παρούσα βιβλιογραφική ανασκόπηση συνοψίζει την υπάρχουσα έρευνα σε αυτόν τον τομέα και επισημαίνει τις προκλήσεις και τις μελλοντικές κατευθύνσεις.

5.6.1 Ανίχνευση με χρήση μηχανικής μάθησης

Μια από τις πρώτες μελέτες για την ανίχνευση πλαστών εικόνων με χρήση μηχανικής μάθησης διεξήχθη από τους Farid και Lyu (2004), οι οποίοι πρότειναν μια στατιστική μέθοδο για την ανίχνευση πλαστών ψηφιακών εικόνων αναλύοντας τις ασυνέπειες στις ιδιότητες της εικόνας. Έκτοτε, έχουν αναπτυχθεί πολλές τεχνικές βασισμένες στη μηχανική μάθηση για την ανίχνευση πλαστών εικόνων, συμπεριλαμβανομένων μοντέλων βαθιάς μάθησης, όπως τα Συνελικτικά Νευρωνικά Δίκτυα (CNN), τα Παραγωγικά Αντιπαραθετικά Δίκτυα (GAN) και τα Επαναλαμβανόμενα Νευρωνικά Δίκτυα (RNN).

5.6.2 Ανίχνευση με χρήση GAN

Το 2017, μια μελέτη των Nguyen et al. (2017) πρότεινε μια μέθοδο βασισμένη σε Παραγωγικά Αντιπαραθετικά Δίκτυα (GAN) για την ανίχνευση πλαστών εικόνων που παράγονται από GANs. Οι συγγραφείς χρησιμοποίησαν μια τροποποιημένη έκδοση της αρχιτεκτονικής GAN για τη δημιουργία συνθετικών εικόνων και στη συνέχεια χρησιμοποίησαν ένα CNN για να ταξινομήσουν τις παραγόμενες εικόνες ως πραγματικές ή ψεύτικες. Τα αποτελέσματα έδειξαν ότι η προτεινόμενη μέθοδος υπερτερεί έναντι άλλων σύγχρονων μεθόδων στην ανίχνευση εικόνων που παράγονται από GAN.

5.6.3 Ανίχνευση με χρήση CNN

Το 2018, μια μελέτη από τους Li κ.ά. (2018) πρότεινε μια μέθοδο βασισμένη σε Συνελικτικά Νευρωνικά Δικτύα (CNN) για την ανίχνευση της συρραφής εικόνων, η οποία περιλαμβάνει τον συνδυασμό δύο ή περισσότερων εικόνων για τη δημιουργία μιας νέας εικόνας. Η προτεινόμενη μέθοδος χρησιμοποίησε ένα CNN για την εκμάθηση των χαρακτηριστικών των αυθεντικών και των συζευγμένων εικόνων και στη συνέχεια χρησιμοποίησε ένα δέντρο απόφασης για την ταξινόμηση των εικόνων ως αυθεντικές ή συζευγμένες.

5.6.4 Ανίχνευση με συνδυασμό CNN και RNN

Μια άλλη μελέτη από τους Li κ.ά. (2019) πρότεινε ένα υβριδικό μοντέλο για την ανίχνευση τόσο της συγκόλλησης όσο και της χειραγώγησης εικόνων χρησιμοποιώντας έναν συνδυασμό CNN και Επαναλαμβανόμενων Νευρωνικών (RNN). Η προτεινόμενη μέθοδος χρησιμοποίησε ένα CNN για την

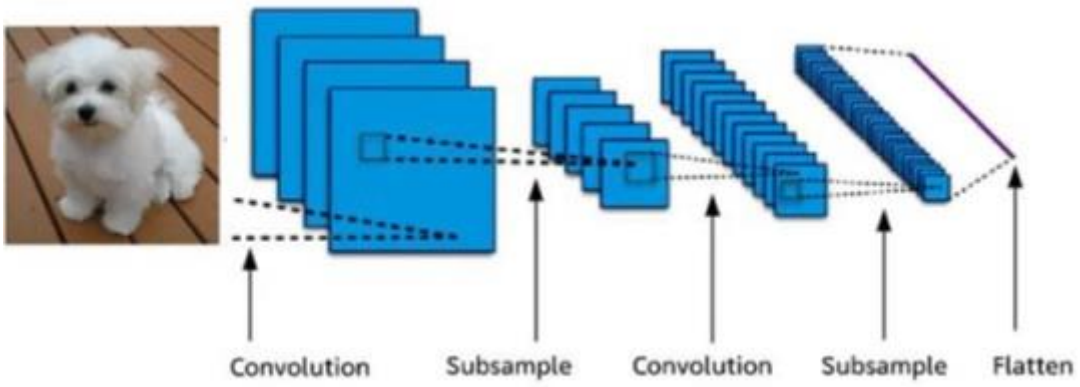
εξαγωγή των χαρακτηριστικών μιας εικόνας και στη συνέχεια χρησιμοποίησε ένα RNN για την ανάλυση των χρονικών εξαρτήσεων μεταξύ των χαρακτηριστικών της εικόνας.

5.6.5 Ανίχνευση με συνδυασμό CNN και GAN

Το 2020, μια μελέτη των Huh et al. (2020) πρότεινε μια μέθοδο για την ανίχνευση βαθιά πλαστών βίντεο χρησιμοποιώντας έναν συνδυασμό CNNs και GANs. Η προτεινόμενη μέθοδος χρησιμοποίησε ένα CNN για την εξαγωγή χαρακτηριστικών από κάθε καρέ ενός βίντεο και στη συνέχεια χρησιμοποίησε ένα GAN για τη δημιουργία ενός ψεύτικου βίντεο. Στη συνέχεια, το παραγόμενο ψεύτικο βίντεο συγκρίθηκε με το αρχικό βίντεο για τον εντοπισμό τυχόν ανακολουθιών.

5.7 Συνελικτικά νευρωνικά δίκτυα

Τα συνελικτικά νευρωνικά δίκτυα χρησιμοποιούνται συχνά στην ανίχνευση πλαστών εικόνων λόγω της ικανότητας τους στην επεξεργασία εικόνων και στην διάκριση των υποκείμενων μοτίβων μέσα σε αυτές. Η διαδικασία ανίχνευσης ψεύτικων εικόνων με την χρήση CNN περιλαμβάνει συνήθως την εκπαίδευση του δικτύου σε ένα μεγάλο σύνολο δεδομένων που περιλαμβάνει τόσο αυθεντικές όσο και πλαστές εικόνες. Στην συνέχεια, το εκπαιδευμένο μοντέλο χρησιμοποιείται για την διάκριση πλαστών εικόνων. Κατά τη διάρκεια της φάσης εκπαίδευσης, το CNN αποκτά την ικανότητα να αναγνωρίζει διακριτά μοτίβα στις εικόνες που διαφοροποιούν τις αυθεντικές από τις πλαστές εικόνες. Τα μοτίβα αυτά μπορεί να περιλαμβάνουν διαφοροποιήσεις στο χρώμα, την υφή ή άλλη οπτικά χαρακτηριστικά που αφορούν αποκλειστικά τις πλαστές εικόνες. Μόλις το CNN εκπαιδευτεί, μπορεί να διακρίνει τις πλαστές εικόνες εισάγοντας μία εικόνα και αναλύοντας την έξοδο που προκύπτει. Συνήθως, η έξοδος του CNN εκφράζεται ως ένα αποτέλεσμα πιθανότητας που υποδηλώνει την περίπτωση η εικόνα να είναι πλαστή. Παρόλα αυτά είναι ζωτικής σημασίας να αναγνωρίσουμε ότι η ανίχνευση πλαστών εικόνων αποτελεί τεράστια πρόκληση και ακόμη και τα μοντέλα CNN προηγμένης τεχνολογίας μπορούν να εξαπατηθούν που εξελιγμένες απομιμήσεις. Κατά συνέπεια, η συνεχής βελτίωση και εξέλιξη αυτών των μεθοδολογιών είναι επιτακτική ανάγκη για να ξεπεράσουμε την εξέλιξη των τεχνικών δημιουργίας πλαστών εικόνων [2].



Εικόνα 5.3: Βασικό διάγραμμα CNN [2]

5.8 Μεθοδολογία παρούσας εργασίας

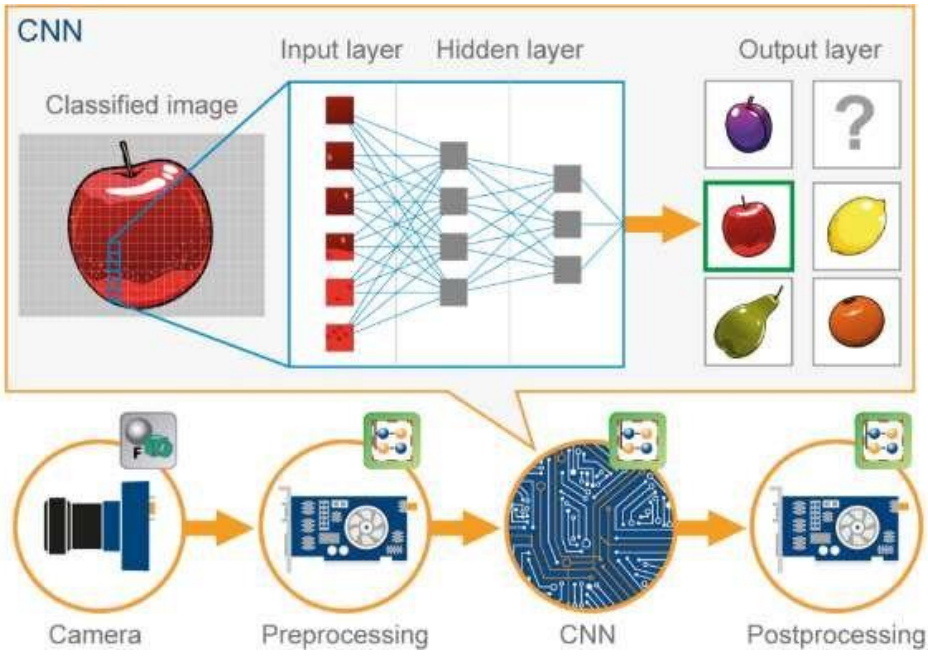
Η ανίχνευση παραποιημένων εικόνων με χρήση μηχανικής εκμάθησης συνεπάγεται την εκπαίδευση ενός μοντέλου για τη διάκριση μεταξύ πραγματικών και πλαστών εικόνων. Ακολουθεί μια γενική επισκόπηση των βημάτων που απαιτούνται για την υλοποίηση ενός τέτοιου μοντέλου.

Αρχικά πρέπει να γίνει η συλλογή και προετοιμασία ενός συνόλου δεδομένων. Θα γίνεται η συλλογή των δεομένων εικόνων, που περιλαμβάνει τόσο πραγματικά όσο και παραποιημένα παραδείγματα. Το σύνολο δεδομένων θα πρέπει να είναι εκτεταμένο και ποικίλο, καλύπτοντας ένα ευρύ φάσμα σεναρίων. Αφού συλλεχτεί το σύνολο δεδομένων χρειάζεται προετοιμασία για την εκπαίδευση του μοντέλου μηχανικής μάθησης. Αυτό συνήθως περιλαμβάνει εργασίες όπως η αλλαγή μεγέθους, η κανονικοποίηση και η προεπεξεργασία εικόνων.

Συνεχίζουμε με τον ορισμό της αρχιτεκτονικής μοντέλου. Πρέπει να καθοριστεί η αρχιτεκτονική του μοντέλου μηχανικής μάθησης που προορίζεται για τον εντοπισμό πλαστών εικόνων. Αυτό μπορεί να επιτευχθεί με την χρήση διάφορων προσεγγίσεων, όπως τα συνελικτικά νευρωνικά δίκτυα (CNN), τα παραγωγικά αντιπαραθετικά δίκτυα (GAN) ή άλλες αρχιτεκτονικές βαθιάς μάθησης.

Μετά ακολουθεί η εκπαίδευση του μοντέλου. Με καθορισμένη την αρχιτεκτονική του μοντέλου, ξεκινάει η εκπαίδευση. Αυτό συνεπάγεται την τροφοδοσία του προετοιμασμένου συνόλου δεδομένων στο μοντέλο και την δυνατότητα να μάθει τα χαρακτηριστικά που διαφοροποιούν τις πραγματικές από τις ψεύτικες εικόνες. Κατά τη διάρκεια της εκπαίδευσης, μπορεί να χρειαστούν προσαρμογές στις υπερπαραμέτρους όπως ο ρυθμός μάθησης, το μέγεθος δέσμης και η κανονικοποίηση για να βελτιστοποιηθεί η απόδοση του μοντέλου.

Τελειώνουμε με την επιβεβαίωση του μοντέλου. Μετά την εκπαίδευση, πρέπει να επικυρωθεί το μοντέλο για να διασφαλιστεί η αποτελεσματικότητα του. Αυτό περιλαμβάνει την δοκιμή του μοντέλου σε ένα ξεχωριστό σύνολο δεδομένων εικόνων στο οποίο δεν έχει εκπαιδευτεί. Η απόδοση του μοντέλου αξιολογείται με τον υπολογισμό μεταβλητών όπως η ακρίβεια.



Εικόνα 5.4: Ταξινόμηση εικόνας [2]

5.9 Επίλογος

Η παρούσα ανασκόπηση μιλάει την ανίχνευση των παραποιημένων εικόνων και την αναγνώριση τους είτε χωρίς είτε με μηχανική μάθηση. Ορίστηκε η μηχανική μάθηση και αναδείχθηκαν δυνατότητες διαφόρων τεχνικών μηχανικής μάθησης, όπως τα CNN, τα GAN και τα RNN, στον εντοπισμό πλαστών εικόνων. Οι πρώιμες μελέτες των Farid και Lyu (2004) έθεσαν τις βάσεις, ενώ οι πρόσφατες προσεγγίσεις χρησιμοποιούν προηγμένα μοντέλα βαθιάς μάθησης. Αξιοσημείωτες εργασίες περιλαμβάνουν την ανίχνευση με βάση το GAN από τους Nguyen κ.ά. (2017), την ανίχνευση συρραφής εικόνων με βάση το CNN από τους Li κ.ά. (2018) και υβριδικά μοντέλα που συνδυάζουν CNN και RNN από τους Li κ.ά. (2019). Έγινε μια αρχική αναφορά στο CNN και παρουσιάστηκε η μεθοδολογία που θα ακολουθήσουμε στην παρούσα εργασία.

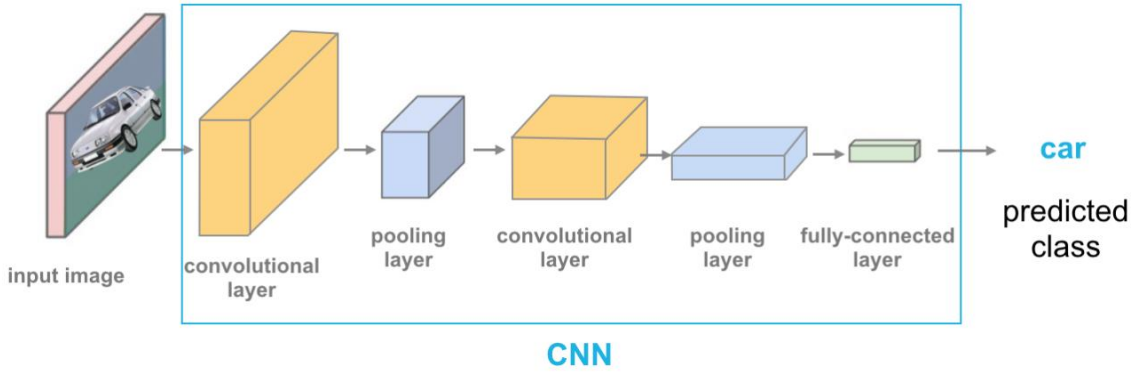
Κεφάλαιο 6ο: Επεξήγηση και υλοποίηση του μοντέλου

6.1 Εισαγωγή

Στην παρούσα πτυχιακή εργασία, στοχεύσαμε στην κατηγοριοποίηση εικόνων. Οι δύο κατηγορίες είναι οι εικόνες με deepfake και οι εικόνες χωρίς deepfake. Η κατηγοριοποίηση αυτή επιτεύχθηκε με την χρήση μηχανικής μάθησης και πιο συγκεκριμένα με τα Συνελικτικά Νευρωνικά Δίκτυα (CNN). Το σύνολο των δεδομένων που χρησιμοποιήθηκαν για την υλοποίηση του μοντέλου είναι το OpenForensics: Large-Scale Challenging Dataset For Multi-Face Forgery Detection And Segmentation In-The-Wild [29]. Παρακάτω θα γίνει ανάλυση των τεχνολογιών που χρησιμοποιήθηκαν, της υλοποίησης του κώδικα και θα παρουσιαστεί το αποτέλεσμα αυτών.

6.2 CNN και αναγνώριση εικόνων

Στον τομέα της αναγνώρισης εικόνων, τα CNN έχουν δείξει αξιοσημείωτη επιτυχία σε εργασίες όπως η αναγνώριση αντικειμένων, η κατανόηση σκηνών και η αναγνώριση προσώπου [30]. Υπερέχουν σε εργασίες αναγνώρισης εικόνας λόγω της ικανότητάς τους να μαθαίνουν αυτόματα ιεραρχικές αναπαραστάσεις οπτικών χαρακτηριστικών απευθείας από τα ακατέργαστα δεδομένα εικονοστοιχείων (pixels) [31]. Τα συνελικτικά στρώματα εντός των CNN λειτουργούν ως ανιχνευτές χαρακτηριστικών, αναγνωρίζοντας μοτίβα και δομές σε διαφορετικά επίπεδα. Για παράδειγμα, τα πρώιμα στρώματα μπορεί να ανιχνεύουν απλά χαρακτηριστικά, όπως ακμές και γωνίες, ενώ τα βαθύτερα στρώματα μαθαίνουν προοδευτικά πιο σύνθετα και αφηρημένα χαρακτηριστικά, όπως μέρη αντικειμένων και υφές [31]. Αυτή η ιεραρχική διαδικασία εξαγωγής χαρακτηριστικών επιτρέπει στα CNN να διακρίνουν περίπλοκα μοτίβα και παραλλαγές εντός των εικόνων, διευκολύνοντας την ακριβή αναγνώριση και ταξινόμηση.



Εικόνα 6.1: Αρχιτεκτονική CNN [32]

Επιπλέον, η ιεραρχική αρχιτεκτονική των CNN αντικατοπτρίζει στενά την ιεραρχική οργάνωση της οπτικής επεξεργασίας στο οπτικό σύστημα των θηλαστικών [31], κάτι που έχει συμβάλει στην αποτελεσματικότητα τους. Μιμούμενοι την ιεραρχική επεξεργασία των οπτικών ερεθισμάτων στον

εγκέφαλο, τα CNN επιδεικνύουν μια εγγενή ικανότητα εξαγωγής ουσιαστικών χαρακτηριστικών από εικόνες, παρόμοια με τον τρόπο με τον οποίο το ανθρώπινο οπτικό σύστημα διακρίνει αντικείμενα και σκηνές.

Η εφαρμογή των CNN στην αναγνώριση εικόνων έχει φέρει επανάσταση σε πολλές βιομηχανίες, όπως η υγειονομική περίθαλψη, η αυτοκινητοβιομηχανία και η ασφάλεια. Στον τομέα της υγειονομικής περίθαλψης, χρησιμοποιούνται για την ανάλυση ιατρικών εικόνων, βοηθώντας στη διάγνωση ασθενειών από ακτινολογικές εικόνες όπως οι ακτίνες Χ και οι μαγνητικές τομογραφίες [31]. Στα αυτόνομα οχήματα, τα συστήματα που βασίζονται σε CNN επιτρέπουν την ισχυρή ανίχνευση και ταξινόμηση αντικειμένων, ενισχύοντας την ασφάλεια και την αποτελεσματικότητα στους δρόμους. Επιπλέον, τα CNN διαδραματίζουν ζωτικό ρόλο σε εφαρμογές ασφάλειας και επιτήρησης, διευκολύνοντας την αναγνώριση και τον εντοπισμό αντικειμένων και ατόμων σε ροές βίντεο σε πραγματικό χρόνο.

Συνολικά, τα CNN είναι ένα ιδιαίτερα χρήσιμο εργαλείο στην αναγνώριση εικόνας, αξιοποιώντας τεχνικές μηχανικής μάθησης για την εξαγωγή και ερμηνεία οπτικών πληροφοριών με πρωτοφανή ακρίβεια και αποτελεσματικότητα.

6.3 Keras και TensorFlow

Το Keras και το TensorFlow είναι σημαντικά εργαλεία στο πεδίο της μηχανικής μάθησης, που χρησιμεύουν τόσο για ερευνητικές όσο και για βιομηχανικές εφαρμογές. Το TensorFlow, που αναπτύχθηκε από την Google Brain, είναι μια βιβλιοθήκη μηχανικής μάθησης ανοικτού κώδικα, γνωστή για την επεκτασιμότητα και την ευελιξία της στην κατασκευή και την εκπαίδευση διαφόρων μοντέλων μηχανικής μάθησης [33]. Προσφέρει ένα ολοκληρωμένο οικοσύστημα για εργασίες που κυμαίνονται από τα νευρωνικά δίκτυα έως την ενισχυτική μάθηση. Το Keras, από την άλλη πλευρά, είναι ένα υψηλού επιπέδου API νευρωνικών δικτύων γραμμένο σε Python, σχεδιασμένο για γρήγορο πειραματισμό και δημιουργία πρωτοτύπων [34]. Παρέχει μια διεπαφή για την κατασκευή νευρωνικών δικτύων και αφαιρεί μεγάλο μέρος της πολυπλοκότητας που εμπλέκεται στην ανάπτυξη μοντέλων μηχανικής μάθησης. Αξίζει να σημειωθεί ότι το Keras μπορεί να εκτελεστεί πάνω από το TensorFlow, πετυχαίνοντας την αξιοποίηση των ισχυρών δυνατοτήτων που προσφέρει το TensorFlow, διατηρώντας παράλληλα τον φιλικό προς τον χρήστη σχεδιασμό του [34]. Αυτή η ενσωμάτωση έχει εδραιώσει το Keras ως μια δημοφιλή επιλογή τόσο μεταξύ των αρχάριων όσο και των έμπειρων επαγγελματιών της μηχανικής μάθησης. Μαζί, το TensorFlow και το Keras δίνουν τη δυνατότητα στους ερευνητές και τους προγραμματιστές να καινοτομούν στον τομέα της τεχνητής νοημοσύνης.

6.4 Δομή του κώδικα

Προκειμένου να πραγματοποιηθεί η κατηγοριοποίηση εικόνων, αναπτύχθηκε κώδικας για να δημιουργηθεί ένα μοντέλο χρησιμοποιώντας τα CNN ως μέθοδο για την εκπαίδευση του. Ο κώδικας χωρίζεται σε 3 μεγάλα μέρη. Το πρώτο μέρος αφορά το φόρτωμα και την προεπεξεργασία των εικόνων, το δεύτερο μέρος αφορά το χτίσιμο του μοντέλου και το τρίτο μέρος αφορά την εκπαίδευση του μοντέλου.

```
from data_loader import load_and_preprocess_data
from model_builder import build_model
from model_trainer import train_model

# Load and preprocess data
X_train, X_val, y_train, y_val = load_and_preprocess_data()

# Build the model
model = build_model()

# Train the model
trained_model, history = train_model(model, X_train, y_train, X_val, y_val)
```

Εικόνα 6.2: Δομή κώδικα

6.5 Φόρτωση και προεπεξεργασία δεδομένων εκπαίδευσης

Στο στάδιο της φόρτωσης και της προεπεξεργασίας των εικόνων, στόχος είναι να προετοιμαστεί το σύνολο των δεδομένων για αποτελεσματική εκπαίδευση. Όπως θα δούμε και αναλυτικότερα παρακάτω, η εκπαίδευση στη μηχανική μάθηση αναφέρεται στην παρουσίαση πολλών προτύπων στο σύστημα με σκοπό την ρύθμιση των παραμέτρων του ώστε αυτό να βελτιώνεται στην λειτουργία αναγνώρισης ή σε όποια άλλη λειτουργία ορίστηκε. Η φόρτωση περιλαμβάνει την πρόσβαση στα ακατέργαστα δεδομένα από την πηγή τους, που στην περίπτωσή μας είναι αρχεία εικόνων από τον δίσκο. Μετά τη φόρτωση, εκτελούνται τα βήματα προεπεξεργασίας των εικόνων που είναι ζωτικής σημασίας για τη διασφάλιση ότι τα δεδομένα είναι σε κατάλληλη μορφή για το μοντέλο. Αυτό συνήθως περιλαμβάνει εργασίες όπως η αλλαγή μεγέθους των εικόνων, η κανονικοποίηση των τιμών των pixels ή η μετατροπή των κατηγορικών μεταβλητών σε αριθμητικές αναπαραστάσεις. Η προεπεξεργασία των δεδομένων είναι απαραίτητη για την ενίσχυση της ικανότητας του μοντέλου να μαθαίνει μοτίβα από τα δεδομένα, αφαιρώντας τον θόρυβο και τις ασυνέπειες.

6.5.1 Εισαγωγή βιβλιοθηκών

Για αρχή εισάγονται οι απαραίτητες βιβλιοθήκες ώστε να πετύχουμε την φόρτωση και την επεξεργασία των δεδομένων. Πιο αναλυτικά, εισάγεται η 'keras' από 'tensorflow' η οποία είναι μια βιβλιοθήκη νευρωνικών δικτύων ανοικτού κώδικα γραμμένη σε Python όπου χρησιμοποιείται για την προεπεξεργασία δεδομένων και τη δημιουργία μοντέλων νευρωνικών δικτύων. Η βιβλιοθήκη 'os' παρέχει λειτουργίες για την αλληλεπίδραση με το λειτουργικό σύστημα επομένως χρησιμοποιείται για το χειρισμό διαδρομών αρχείων και καταλόγων. Η βιβλιοθήκη 'cv2' είναι μια δημοφιλής βιβλιοθήκη για εργασίες μηχανικής όρασης και επεξεργασίας εικόνας όπου χρησιμοποιείται για την ανάγνωση αρχείων εικόνας. Η βιβλιοθήκη 'NumPy' είναι ένα πακέτο για αριθμητικούς υπολογισμούς στην Python όπου παρέχει υποστήριξη για μεγάλους, πολυδιάστατους πίνακες, μαζί με μια συλλογή μαθηματικών συναρτήσεων για τη λειτουργία αυτών των πινάκων. Η βιβλιοθήκη 'Image' από 'PIL' προσθέτει υποστήριξη για το άνοιγμα, τον χειρισμό και την αποθήκευση πολλών διαφορετικών μορφών αρχείων εικόνας επομένως χρησιμοποιείται για την αλλαγή του μεγέθους των εικόνων. Η βιβλιοθήκη 'train_test_split' από 'sklearn.model_selection' χρησιμοποιείται για το διαχωρισμό του συνόλου

δεδομένων σε σύνολα εκπαίδευσης και επικύρωσης, ανακατεύοντας τα τυχαία και τα χωρίζοντας τα σε δύο τμήματα σύμφωνα με το καθορισμένο μέγεθος δοκιμής. Τέλος, η config είναι μια κλάση φτιαγμένη από εμάς που περιέχει διάφορες σταθερές και παραμέτρους που χρησιμοποιούνται σε όλο το έργο. Βοηθά στη διατήρηση της συνοχής και διευκολύνει την εύκολη προσαρμογή των διαφόρων ρυθμίσεων.

```
import os
import cv2
import numpy as np
from PIL import Image
from sklearn.model_selection import train_test_split
from tensorflow import keras
from config import Config
```

Εικόνα 6.3: Απαραίτητες βιβλιοθήκες

6.5.2 Προκαθορισμένες τιμές

Σε αυτό το σημείο θα δούμε τις προκαθορισμένες τιμές που θα χρησιμοποιηθούν σε αυτό το μέρος.

Για αρχή ορίζεται η διαδρομή όπου υπάρχουν τα δεδομένα εικόνων τα οποία θα χρησιμοποιηθούν κατά την εκπαίδευση και κατά την δοκιμή του μοντέλου. Αποθηκεύονται αντίστοιχα στις σταθερές TRAIN_PATH και TEST_PATH.

Στην συνέχεια, ορίζεται το ύψος και το πλάτος της εικόνας σε pixels, επομένως καθορίζεται ο τελικός αριθμός των pixels που θα έχει η κάθε εικόνα. Η αλλαγή μεγέθους των εικόνων είναι ένα κρίσιμο βήμα προεπεξεργασίας που συμβάλλει στη βελτίωση της αποδοτικότητας, της γενίκευσης και της απόδοσης των μοντέλων CNN για εργασίες πρόβλεψης εικόνων. Πιο συγκεκριμένα διασφαλίζει ότι όλες οι εικόνες που τροφοδοτούνται στο δίκτυο έχουν τις ίδιες διαστάσεις, κάτι που είναι απαραίτητο για την αποτελεσματική επεξεργασία παρτίδων και τις λειτουργίες πινάκων. Μειώνει την υπολογιστική πολυπλοκότητα του μοντέλου μιας και οι μικρότερες εικόνες απαιτούν λιγότερες παραμέτρους και λιγότερη μνήμη. Βελτιώνει την ικανότητα γενίκευσης του μοντέλου μιας και με τη μείωση της χωρικής ανάλυσης των εικόνων, αυτό γίνεται πιο ανθεκτικό στις μεταβολές του μεγέθους και της κλίμακας εισόδου, κάτι που εμφανίζεται στα σενάρια του πραγματικού κόσμου. Με τον όρο γενίκευση (generalization) στην μηχανική μάθηση αναφερόμαστε στην ικανότητα να εκτιμάμε τη σωστή έξοδο για πρότυπα εισόδου που δεν έχουμε δει κατά την εκπαίδευση [35]. Τέλος, λόγω των μικρότερων εικόνων πετυχαίνουμε ταχύτερους χρόνους εκπαίδευσης και εξαγωγής συμπερασμάτων, επιτρέποντας το γρήγορο πειραματισμό του μοντέλου. Η επιλογή του μεγέθους των εικόνων δεν είναι εύκολη διαδικασία, μιας και αποτελεί συμβιβασμό μεταξύ διαφόρων παραγόντων. Ένας μεγάλος αριθμός pixel προσφέρει καλύτερη διατήρηση των λεπτομερειών της εικόνας, καλύτερη αναπαράσταση χαρακτηριστικών και καλύτερη απόδοση σε σύνθετες εργασίες. Από την άλλη, οι μεγάλες εικόνες απαιτούν περισσότερους υπολογιστικούς πόρους και τα μοντέλα που εκπαιδεύονται σε μεγαλύτερες εικόνες ενδέχεται να είναι πιο επιρρεπή σε υπερπροσαρμογή ιδίως εάν το σύνολο δεδομένων δεν είναι αρκετά μεγάλο. Στην μηχανική μάθηση, η υπερπροσαρμογή (overfitting) συμβαίνει όταν το μοντέλο γίνεται περίπλοκο και απομνημονεύει τα δεδομένα εκπαίδευσης, με αποτέλεσμα την καλή ακρίβεια στα δεδομένα εκπαίδευσης αλλά την κακή γενίκευση σε νέα, αθέατα δεδομένα [36]. Για το δικό μας μοντέλο, μετά από δοκιμές, επιλέξαμε τις διαστάσεις 128 για το ύψος και 128 για το πλάτος. Θεωρούμε

ότι αυτές οι τιμές προσφέρουν την ισορροπία μεταξύ της διατήρησης της λεπτομέρειας και της υπολογιστικής αποδοτικότητας επιτυγχάνοντας έτσι το επιθυμητό τελικό αποτέλεσμα. Οι τιμές αποθηκεύονται στις σταθερές IMG_HEIGHT και IMG_WIDTH.

Τέλος, ορίζεται ο αριθμός των διαφορετικών κλάσεων ή κατηγοριών όπου θα χρησιμοποιηθούν κατά την εκπαίδευση του μοντέλου. Στη μηχανική μάθηση, ο όρος κλάσεις αναφέρεται στις διακριτές κατηγορίες που στοχεύουμε να ταξινομήσουμε ή να προβλέψουμε σε ένα συγκεκριμένο σύνολο δεδομένων. Αυτές οι κλάσεις αντιπροσωπεύουν διαφορετικά αποτελέσματα, ετικέτες ή καταστάσεις που θέλουμε το μοντέλο μας να μάθει να αναγνωρίζει ή να διακρίνει μεταξύ τους [37]. Οι κλάσεις χρησιμεύουν ως οι ετικέτες αληθείας που παρέχονται κατά τη διάρκεια της εκπαίδευσης όπου το μοντέλο μαθαίνει να αντιστοιχίζει τα χαρακτηριστικά εισόδου σε αυτές τις προκαθορισμένες κλάσεις. Κατά τη διάρκεια της εκπαίδευσης, το μοντέλο προσαρμόζει τις παραμέτρους του ώστε να ελαχιστοποιήσει την απόκλιση μεταξύ των προβλεπόμενων και των πραγματικών ετικετών κλάσεων. Οι κλάσεις μπορεί να είναι είτε διακριτές είτε συνεχείς. Οι διακριτές κλάσεις, όπως και στην περίπτωσή μας, αντιπροσωπεύουν διακριτές κατηγορίες (π.χ. είδη ζώων), ενώ οι συνεχείς κλάσεις αντιπροσωπεύουν ένα εύρος τιμών (π.χ. τιμή ενός σπιτιού). Οι εργασίες ταξινόμησης συνήθως ασχολούνται με διακριτές κλάσεις, ενώ οι εργασίες παλινδρόμησης με συνεχείς κλάσεις. Έτσι, η κατανόηση των κλάσεων είναι θεμελιώδης στην επιβλεπόμενη μάθηση, όπου το μοντέλο μαθαίνει από δεδομένα με ετικέτες ώστε να φτάσει στο σημείο να κάνει προβλέψεις σε νέα, αθέατα δεδομένα. Εφόσον έχουμε σταθερό και γνωστό αριθμό κλάσεων στα δεδομένα μας, στον κώδικα ορίζουμε την τιμή 2 σαν αριθμό κλάσεων. Μια κλάση αντιστοιχεί στις αυθεντικές φωτογραφίες και η άλλη στις παραποιημένες φωτογραφίες.

```
DATA_DIR = "C[redacted]/Desktop/Dataset"
TRAIN_PATH = os.path.join(DATA_DIR, "Train")
IMG_HEIGHT, IMG_WIDTH = 128, 128
NUM_CATEGORIES = 2
```

Εικόνα 6.4: Σταθερές της κλάσης

6.5.3 Φόρτωση δεδομένων εκπαίδευσης

Η μέθοδος load_and_preprocess_data() ξεκινά με την αρχικοποίηση δύο κενών λιστών, την image_data για την αποθήκευση των δεδομένων εικόνας και την image_labels για την αποθήκευση των αντίστοιχων ετικετών. Στην συνέχεια, θα χρησιμοποιηθεί ένας βρόγχος επανάληψης για κάθε κατηγορία. Ο αριθμός των κατηγοριών καθορίζεται από το Config.NUM_CATEGORIES που έχει οριστεί σε 2, όπου είναι και ο συνολικός αριθμός των κατηγοριών στο σύνολο δεδομένων. Για κάθε κατηγορία, ορίζεται η διαδρομή προς τον αντίστοιχο κατάλογο που περιέχει τις εικόνες εκπαίδευσης χρησιμοποιώντας το os.path.join(Config.TRAIN_PATH, str(i)). Εντός του εμφωλευμένου βρόγχου η συνάρτηση διαβάζει κάθε αρχείο εικόνας από τον τρέχοντα κατάλογο, χρησιμοποιώντας το cv2.imread(os.path.join(path, img)). Στη συνέχεια, με το Image.fromarray(image, 'RGB') μετατρέπει τον NumPy πίνακα σε ένα PIL ‘Image’ αντικείμενο, επιτρέποντας έτσι λειτουργίες επεξεργασίας εικόνας που βασίζονται στην PIL. Το 'RGB' υποδεικνύει ότι η εικόνα θα είναι στο χρωματικό χώρο RGB, ο οποίος αποτελείται από τρία χρωματικά κανάλια, το κόκκινο, πράσινο και μπλε. Στην συνέχεια αλλάζει το μέγεθός των εικόνων στις καθορισμένες διαστάσεις (128x128) χρησιμοποιώντας το

image_fromarray.resize((Config.IMG_HEIGHT, Config.IMG_WIDTH)), διασφαλίζοντας έτσι ότι όλες οι εικόνες θα έχουν σταθερό μέγεθος για την εκπαίδευση. Έπειτα, με το image_data.append(np.array(resize_image)) η εικόνα με τις νέες διαστάσεις γίνεται ξανά NumPy πίνακας και μπαίνει στην 'image_data' λίστα που πλέον θα περιέχει την εικόνα με την νέα διάσταση. Με το image_labels.append(i) θα αντιστοιχηθεί η τρέχουσα εικόνα με την αντίστοιχή ετικέτα της. Τέλος, χρησιμοποιείται ένας μηχανισμός ώστε να θα διαχειριστεί κάποιο σφάλμα σε περίπτωση που προκύψει κατά την παραπάνω διαδικασία. Σε αυτήν την περίπτωση εκτυπώνει ένα μήνυμα που αναφέρει το όνομα του αρχείου (img) και το σφάλμα (ex). Αυτό βοηθά στην αποσφαλμάτωση τυχόν προβλημάτων που παρουσιάζονται κατά την επεξεργασία εικόνας.

```
# Collection of training data
image_data = []
image_labels = []

for i in range(Config.NUM_CATEGORIES):
    path = os.path.join(Config.TRAIN_PATH, str(i))
    images = os.listdir(path)

    for img in images:
        try:
            image = cv2.imread(os.path.join(path, img))
            image_fromarray = Image.fromarray(image, 'RGB')
            resize_image = image_fromarray.resize((Config.IMG_HEIGHT, Config.IMG_WIDTH))
            image_data.append(np.array(resize_image))
            image_labels.append(i)
        except Exception as ex:
            print(f"Error processing {img}: {ex}")
```

Εικόνα 6.5: Συλλογή δεδομένων εκπαίδευσης

6.5.4 Μετατροπή της λίστας σε NumPy πίνακα

Μόλις ολοκληρωθεί η επεξεργασία όλων των εικόνων όλων των κατηγοριών, οι λίστες μετατρέπονται σε πίνακες NumPy με τη χρήση των np.array(image_data) και np.array(image_labels) για την αποτελεσματικότερη αποθήκευση και χειρισμό τους. Μέσα στον βρόγχο είχε μετατραπεί κάθε εικόνα ξεχωριστά σε πίνακα NumPy ενώ εδώ μετατρέπονται ολόκληρες οι λίστες εικόνων και ετικετών σε πίνακες NumPy, δημιουργώντας μια πιο δομημένη και ομοιογενή αναπαράσταση του συνόλου δεδομένων.

```
# Convert the list to a NumPy array
image_data = np.array(image_data)
image_labels = np.array(image_labels)
```

Εικόνα 6.6: Μετατροπή λίστας σε NumPy πίνακα

6.5.5 Ανακάτεμα δεδομένων εκπαίδευσης

Σε αυτό το σημείο πραγματοποιείται μια διαδικασία που ονομάζεται ανακάτεμα δεδομένων και είναι ένα κρίσιμο βήμα στην εκπαίδευση μοντέλων μηχανικής μάθησης, ιδίως νευρωνικών δικτύων όπως τα CNN. Συγκεκριμένα, το ανακάτεμα των δεδομένων βοηθά στην αποφυγή προκαταλήψεων που μπορεί να προκύψουν εάν τα δεδομένα παρουσιαστούν στο μοντέλο με συγκεκριμένη σειρά. Για παράδειγμα, εάν τα δεδομένα ταξινομηθούν με βάση την ετικέτα της κλάσης και εισαχθούν στο μοντέλο, το μοντέλο μπορεί να μάθει μοτίβα που σχετίζονται με τη σειρά των κλάσεων και όχι με τα πραγματικά χαρακτηριστικά. Επίσης, το ανακάτεμα των δεδομένων εξασφαλίζει ότι το μοντέλο βλέπει ένα ποικίλο φάσμα παραδειγμάτων κατά τη διάρκεια κάθε εποχής εκπαίδευσης (ένα πέρασμα από όλα τα δεδομένα εκπαίδευσης), το οποίο μπορεί να βοηθήσει στην αποφυγή της υπερπροσαρμογής, μιας και δίνει στο μοντέλο διαφορετικές παραλλαγές των δεδομένων. Ακόμα, παρουσιάζοντας τα δεδομένα με τυχαία σειρά, το μοντέλο μαθαίνει να γενικεύει καλύτερα μιας και αναγκάζεται να μάθει τα πραγματικά χρήσιμα μοτίβα στα δεδομένα αντί να απομνημονεύει συγκεκριμένες ακολουθίες [38]. Στον κώδικα, αρχικά δημιουργείται ο πίνακας shuffle_indexes με δείκτες που κυμαίνονται από το 0 έως τον αριθμό των δειγμάτων στο σύνολο δεδομένων. Αυτός ο πίνακας αντιπροσωπεύει τους δείκτες των δειγμάτων στο σύνολο δεδομένων. Στη συνέχεια, οι δείκτες ανακατεύονται τυχαία χρησιμοποιώντας τη συνάρτηση np.random.shuffle(). Αυτό το τυχαίο ανακάτεμα εξασφαλίζει ότι τα δείγματα αναδιατάσσονται με τυχαία σειρά. Τέλος, τόσο τα δεδομένα εικόνας όσο και οι αντίστοιχες ετικέτες τους αναδιατάσσονται με βάση τους ανακατεμένους δείκτες. Τα δεδομένα εικόνας και οι ετικέτες αναδιατάσσονται με τέτοιο τρόπο ώστε να αντιστοιχούν η μία στη νέα σειρά της άλλης, ανακατεύοντας ουσιαστικά τις γραμμές του πίνακα δεδομένων εικόνας και τα στοιχεία του πίνακα ετικετών ταυτόχρονα.

```
# Shuffle the training data
shuffle_indexes = np.arange(image_data.shape[0])
np.random.shuffle(shuffle_indexes)
image_data = image_data[shuffle_indexes]
image_labels = image_labels[shuffle_indexes]
```

Εικόνα 6.7: Ανακάτεμα δεδομένων εκπαίδευσης

6.5.6 Διαχωρισμός των δεδομένα σε σύνολα εκπαίδευσης και επικύρωσης

Το σύνολο εκπαίδευσης και το σύνολο επικύρωσης είναι βασικά στοιχεία στη διαδικασία εκπαίδευσης μοντέλων μηχανικής μάθησης, συμπεριλαμβανομένων των νευρωνικών δικτύων συνελικτικής μάθησης. Διαδραματίζουν κρίσιμο ρόλο στην αξιολόγηση της απόδοσης του μοντέλου, στην αποφυγή της υπερπροσαρμογής και στη βελτιστοποίηση των δυνατοτήτων γενίκευσης του.

Το σύνολο εκπαίδευσης χρησιμοποιείται για την εκπαίδευση του μοντέλου CNN. Αποτελείται από επισημειωμένα δείγματα δεδομένων (εικόνες στην προκειμένη περίπτωση) μαζί με τις αντίστοιχες ετικέτες τους. Κατά τη διάρκεια της εκπαίδευσης, το μοντέλο μαθαίνει να αναγνωρίζει μοτίβα και χαρακτηριστικά στα δεδομένα και να προσαρμόζει τις εσωτερικές του παραμέτρους, γνωστές ως βάρη και προκαταλήψεις, ώστε να ελαχιστοποιεί την απόκλιση μεταξύ των προβλεπόμενων εξόδων και των ετικετών [39]. Έτσι, το μοντέλο μαθαίνει από τα διαθέσιμα δεδομένα και βελτιώνει την απόδοσή του

με την πάροδο του χρόνου. Παρέχει τις απαραίτητες πληροφορίες ώστε το μοντέλο να γενικεύσει τις γνώσεις του και να κάνει ακριβείς προβλέψεις σε νέα δεδομένα.

Το σύνολο επικύρωσης χρησιμοποιείται για την αξιολόγηση της απόδοσης του εκπαιδευμένου μοντέλου κατά τη διάρκεια της εκπαίδευσης. Χρησιμεύει ως ένα ανεξάρτητο σύνολο δεδομένων που βοηθά στην αξιολόγηση της ικανότητας του μοντέλου να γενικεύει σε νέα δεδομένα και στον εντοπισμό πιθανών προβλημάτων, όπως η υπερπροσαρμογή ή η υπεραπλούστευση [39]. Η υπεραπλούστευση (underfitting) συμβαίνει όταν ένα μοντέλο είναι πολύ απλό για να συλλάβει τα υποκείμενα μοτίβα στα δεδομένα, οδηγώντας σε κακή απόδοση τόσο στα σύνολα εκπαίδευσης όσο και στα νέα δεδομένα [36]. Αξιολογώντας τις επιδόσεις του μοντέλου στο σύνολο επικύρωσης, μπορούμε να πάρουμε αποφάσεις σχετικά με την αρχιτεκτονική του μοντέλου, τις παραμέτρους και τη διαδικασία εκπαίδευσης. Αυτή η επαναληπτική διαδικασία επικύρωσης επιτρέπει τη λεπτομερή ρύθμιση του μοντέλου και τη βελτιστοποίηση της απόδοσής του.

Η επιλογή του μεγέθους του συνόλου επικύρωσης είναι ζωτικής σημασίας για την αποτελεσματική αξιολόγηση του μοντέλου. Με μικρό σύνολο επικύρωσης πετυχαίνεται ταχύτερη διαδικασία εκπαίδευσης και αξιολόγησης, κατάλληλη για μεγάλα σύνολα δεδομένων ή περιορισμένους υπολογιστικούς πόρους αλλά μπορεί να οδηγήσει σε λιγότερο αξιόπιστη εκτίμηση της απόδοσης και αυξημένο κίνδυνο υπερπροσαρμογής λόγω περιορισμένων δεομένων επικύρωσης. Η βέλτιστη αναλογία διαχωρισμού εξαρτάται από διάφορους παράγοντες όπως το μέγεθος του συνόλου δεδομένων, η πολυπλοκότητα του μοντέλου και οι διαθέσιμοι υπολογιστικοί πόροι. Στην περίπτωσή μας εχουμε επιλέξει να κατανέμουμε το 30% των δεδομένων στο σύνολο επικύρωσης, απόφαση που πάρθηκε μετά από πειραματισμούς.

Στον κώδικα, η μέθοδος train_test_split() χωρίζει τα δεδομένα σε σύνολα εκπαίδευσης και επικύρωσης χρησιμοποιώντας τις ακόλουθες παραμέτρους: Τα image_data και image_labels, δηλαδή τις εικόνες μαζί με τις αντίστοιχες ετικέτες τους. Την παράμετρο test_size=0.3, όπου ορίζει το ποσοστό του συνόλου δεδομένων που θα συμπεριληφθεί στο σύνολο επικύρωσης (30% στην περίπτωσή μας). Την παράμετρο random_state=2, όπου ορίζει ένα συγκεκριμένο σημείο εκκίνησης για την τυχαιότητα που χρησιμοποιείται στη διαδικασία διαχωρισμού των δεδομένων. Να σημειωθεί ότι δεν έχει σημασία ποιος αριθμός θα επιλεγεί, αρκεί να είναι κάθε φορά ο ίδιος. Τέλος, την παράμετρο shuffle=True που υποδεικνύει ότι θα γίνει ανακάτεμα των δεδομένων πριν από τη διάσπαση, γεγονός που βοηθά στην αποφυγή μεροληψιών στην κατανομή των δεδομένων. Στο τέλος παίρνουμε τους πίνακες X_train, X_val, y_train, και y_val. Σχετικά με το όνομα, να σημειωθεί ότι στον τομέα της μηχανικής μάθησης, συνηθίζεται να χρησιμοποιείται το κεφαλαίο ‘X’ για την αναπαράσταση του πίνακα που περιέχει τα δεδομένα εισόδου, και το πεζό ‘y’ για τις ετικέτες που σχετίζονται με τα δεδομένα εισόδου. Ξεχωρίζουμε έτσι καλύτερα τα χαρακτηριστικά εισόδου και των ετικετών τους, βελτιώνοντας την αναγνωσιμότητα και τη σαφήνεια του κώδικα.

```
# Split the data into training and validation sets
X_train, X_val, y_train, y_val = train_test_split(
    image_data, image_labels, test_size=0.3, random_state=2, shuffle=True
)
```

Εικόνα 6.8: Διαχωρισμός σε σύνολα εκπαίδευσης και επικύρωσης

6.5.7 Κανονικοποίηση δεδομένων

Η κανονικοποίηση δεδομένων είναι ένα ακόμα κρίσιμο βήμα προεπεξεργασίας στη μηχανική μάθηση, εξασφαλίζοντας ότι τα χαρακτηριστικά εισόδου έχουν ομοιόμορφη κλίμακα. Στο πλαίσιο των εικόνων, οι τιμές των pixels κανονικοποιούνται συνήθως στο εύρος [0, 1]. Αυτό επιτυγχάνεται διαιρώντας κάθε τιμή του pixel με τη μέγιστη τιμή (255). Η κανονικοποίηση σταθεροποιεί την εκπαίδευση, και αποτρέπει τα χαρακτηριστικά με μεγάλη κλίμακα να κυριαρχούν στη διαδικασία μάθησης.

```
# Normalize the pixel values to the range [0, 1]
X_train = X_train / 255
X_val = X_val / 255
```

Εικόνα 6.9: Κανονικοποίηση δεδομένων

6.5.8 Κωδικοποίηση ετικετών

Σε αυτό το σημείο η συνάρτηση εκτελεί μια κωδικοποίηση στις ετικέτες της εικόνας χρησιμοποιώντας το keras.utils.to_categorical. Πιο συγκεκριμένα, μετατρέπει τις ακέραιες ετικέτες σε δυαδικά διανύσματα, όπου κάθε διάνυσμα αντιπροσωπεύει την παρουσία ή την απουσία μιας κλάσης. Η κωδικοποίηση είναι μια τυπική πρακτική για εργασίες ταξινόμησης και εξασφαλίζει τη συμβατότητα με μοντέλα νευρωνικών δικτύων. Τα y_train και y_val που δέχεται σαν παράμετρο είναι οι αρχικές ετικέτες του συνόλου εκπαίδευσης και το Config.NUM_CATEGORIES είναι ο αριθμός των κλάσεων, όπου καθορίζει το μήκος των κωδικοποιημένων διανυσμάτων.

```
# Encode the labels
y_train = keras.utils.to_categorical(y_train, Config.NUM_CATEGORIES)
y_val = keras.utils.to_categorical(y_val, Config.NUM_CATEGORIES)
```

Εικόνα 6.10: Κωδικοποίηση ετικετών

6.5.9 Επιστροφή τιμών

Τέλος, η συνάρτηση επιστρέφει τέσσερις NumPy πίνακες, τους X_train, X_val, y_train, και y_val που αντιπροσωπεύουν τα προεπεξεργασμένα δεδομένα εκπαίδευσης και επικύρωσης μαζί με τις αντίστοιχες ετικέτες τους, έτοιμα για χρήση στην εκπαίδευση του μοντέλου. Οι τιμές αυτές θα χρησιμοποιηθούν αργότερα στην μέθοδο train_model(model, X_train, y_train, X_val, y_val).

```
return X_train, X_val, y_train, y_val
```

Εικόνα 6.11: Επιστρεφόμενα προεπεξεργασμένα δεδομένα

6.6 Κατασκευή CNN μοντέλου

Προχωρώντας στην κατασκευή του μοντέλου, το στάδιο αυτό περιλαμβάνει τον καθορισμό της αρχιτεκτονικής και των παραμέτρων του νευρωνικού δικτύου. Οι επιλογές σχεδιασμού, όπως ο αριθμός των στρωμάτων, ο τύπος των στρωμάτων, για παράδειγμα συνελικτικά (convolutional), υποδειγματοληψίας (pooling), πλήρως συνδεδεμένα (fully connected) και οι συναρτήσεις ενεργοποίησης επηρεάζουν σημαντικά την ικανότητα του μοντέλου να μαθαίνει και να γενικεύει από τα δεδομένα. Επιπλέον, κατά την διάρκεια αυτής της φάσης ρυθμίζονται οι αλγόριθμοι αρχικοποίησης παραμέτρων και βελτιστοποίησης, οι οποίοι καθορίζουν τον τρόπο με τον οποίο το μοντέλο μαθαίνει από τα δεδομένα εκπαίδευσης. Ο στόχος κατά τη δημιουργία του μοντέλου, είναι η δημιουργία μιας δομής που είναι ικανή να εντοπίζει μοτίβα και σχέσεις εντός του συνόλου δεδομένων, ενώ παράλληλα είναι υπολογιστικά αποδοτική.

Η επιλογή του σωστού αριθμού των στρωμάτων που θα χρησιμοποιηθούν είναι αρκετά κρίσιμη διαδικασία μιας και ένας μικρός αριθμός στρωμάτων μπορεί να οδηγήσει σε μείωση των δυνατοτήτων του μοντέλου και στην υπεραπλούστευση ενώ ένας μεγάλος αριθμός στρωμάτων μπορεί να οδηγήσει σε πολύ περίπλοκο μοντέλο, ανάγκη για περισσότερους υπολογιστικούς πόρους και υπερπροσαρμογή.

6.6.1 Εισαγωγή βιβλιοθηκών

Όπως έχει αναφερθεί η βιβλιοθήκη 'keras' από 'tensorflow' είναι μια βιβλιοθήκη νευρωνικών δικτυών ανοικτού κώδικα γραμμένη σε Python όπου χρησιμοποιείται για την προεπεξεργασία δεδομένων και τη δημιουργία μοντέλων νευρωνικών δικτύων και η Config είναι μια κλάση, φτιαγμένη από εμάς που περιέχει διάφορες σταθερές και παραμέτρους που χρησιμοποιούνται σε όλο το έργο.

```
from tensorflow import keras
from config import Config
```

Εικόνα 6.12: Απαραίτητες βιβλιοθήκες

6.6.2 Προκαθορισμένες τιμές

Σε αυτό το σημείο θα δούμε τις προκαθορισμένες τιμές που θα χρησιμοποιηθούν σε αυτό το μέρος. Έχουν αναλυθεί ήδη οι τιμές για τα NUM_CATEGORIES, IMG_HEIGHT, IMG_WIDTH επομένως θα εστιάσουμε στο CHANNELS. Στην σταθερά CHANNELS δίνεται η τιμή 3 όπου αντιπροσωπεύει τον αριθμό των καναλιών στην εικόνα εισόδου. Συνήθως, στις έγχρωμες εικόνες ο αριθμός των καναλιών είναι 3 (κόκκινο, πράσινο, μπλε) και αυτό επιλέχτηκε και εδώ.

```
IMG_HEIGHT, IMG_WIDTH, CHANNELS = 128, 128, 3
NUM_CATEGORIES = 2
```

Εικόνα 6.13: Σταθερές της κλάσης

6.6.3 Κατασκευή μοντέλου

Παρακάτω θα παρουσιαστούν τα διάφορα στρώματα που χρησιμοποιήθηκαν για την κατασκευή του μοντέλου.

Ξεκινάμε με τα επίπεδα Συνέλιξης. Τα συνελικτικά στρώματα (Conv2D) είναι τα θεμελιώδη δομικά στοιχεία των CNN. Αποτελούνται από φίλτρα που συνελίσσονται με την εικόνα εισόδου για την εξαγωγή χαρακτηριστικών όπως ακμές, υφές και μοτίβα. Κάθε φίλτρο μαθαίνει να ανιχνεύει συγκεκριμένα χαρακτηριστικά, ολισθαίνοντας στην εικόνα εισόδου και εκτελώντας στοιχειομετρικούς πολλαπλασιασμούς ακολουθούμενους από αθροίσματα. Με το όρισμα filters δίνονται διάφοροι αριθμοί φίλτρων, όπου κάθε φίλτρο μαθαίνει διαφορετικά χαρακτηριστικά. Στο όρισμα kernel_size δίνεται το μέγεθος του πυρήνα συνελικτικής επεξεργασίας. Με το activation='relu' ορίζεται η συνάρτηση ενεργοποίησης που χρησιμοποιείται μετά τη λειτουργία της συνέλιξης. Τέλος, με το input_shape() δίνεται το σχήμα των δεδομένων εισόδου, στην περίπτωσή μας ένας τρισδιάστατος τανυστής που αναπαριστά μια εικόνα με ύψος, πλάτος και RGB κανάλια.

```
keras.layers.Conv2D(filters=16, kernel_size=(3, 3), activation='relu',
                    input_shape=(Config.IMG_HEIGHT, Config.IMG_WIDTH, Config.CHANNELS)),
```

Εικόνα 6.14: Παράδειγμα επιπέδου συνέλιξης

Στην συνέχεια, χρησιμοποιήθηκαν τα επίπεδα Κανονικοποίησης Δέσμης. Σε αυτό το επίπεδο, με το BatchNormalization() προστίθεται ένα στρώμα κανονικοποίησης δέσμης στο μοντέλο. Βοηθά στη βελτίωση της ταχύτητας, της απόδοσης και της σταθερότητας των νευρωνικών δικτύων.

```
keras.layers.BatchNormalization(),
```

Εικόνα 6.15: Παράδειγμα επιπέδου κανονικοποίησης δέσμης

Συνεχίζουμε με τα επίπεδα Υποδειγματοληψίας (MaxPool2D). Σε αυτό το επίπεδο, μειώνονται οι διαστάσεις των δεδομένων εισόδου, λαμβάνοντας τη μέγιστη τιμή από κάθε τμήμα του χάρτη χαρακτηριστικών. Με το pool_size το μέγεθος αυτό ορίζεται σε 4x4.

```
keras.layers.MaxPool2D(pool_size=(4, 4)),
```

Εικόνα 6.16: Παράδειγμα επιπέδου υποδειγματοληψίας

Χρησιμοποιείται ακόμα το επίπεδο Πλήρωσης. Σε αυτό το επίπεδο, με το Flatten() μετατρέπονται τα πολυδιάστατα δεδομένα σε ένα μονοδιάστατο διάνυσμα. Το επίπεδο αυτό χρησιμοποιείται για την μετάβαση από τα συνελικτικά στρώματα σε πλήρως συνδεδεμένα στρώματα.

```
keras.layers.Flatten(),
```

Εικόνα 6.17: Παράδειγμα επιπέδου πλήρωσης

Στην συνέχεια, έχουμε τα Πλήρη Συνδεδεμένα επίπεδα. Σε αυτό το επίπεδο, κάθε νευρώνας σε ένα επίπεδο συνδέεται με κάθε νευρώνα στο επόμενο επίπεδο. Αυτό γίνεται με το keras.layers.Dense(512, activation='relu') όπου το 512 ορίζει τον αριθμό των νευρώνων στο επίπεδο και το 'relu' ορίζει τη συνάρτηση ενεργοποίησης που θα χρησιμοποιηθεί.

```
keras.layers.Dense(512, activation='relu'),
```

Εικόνα 6.18: Παράδειγμα Πλήρες Συνδεδεμένου επιπέδου

Σημαντικό είναι και το επίπεδο Εξασθένησης. Σε αυτό το επίπεδο, με το Dropout(rate=0.5) μηδενίζονται οι κόμβοι του νευρωνικού δικτύου κατά τη διάρκεια της εκπαίδευσης για να αποτρέψουν την υπερπροσαρμογή. Με το 0.5 ορίζουμε ότι το 50% των νευρώνων θα μηδενιστούν.

```
keras.layers.Dropout(rate=0.5),
```

Εικόνα 6.19: Παράδειγμα επιπέδου εξασθένησης

Τέλος, έχουμε το επίπεδο Εξόδου. Το επίπεδο εξόδου παράγει προβλέψεις με βάση την είσοδο που λαμβάνει από τα προηγούμενα στρώματα. Με το Dense(Config.NUM_CATEGORIES, activation='softmax') ορίζεται ο αριθμός των κλάσεων, που είναι 2 στην περίπτωσή μας, και η συνάρτηση ενεργοποίησης softmax όπου έχει σαν έξοδο μια κατανομή πιθανοτήτων πάνω στις κλάσεις, εξασφαλίζοντας ότι το άθροισμα των πιθανοτήτων είναι 1.

```
keras.layers.Dense(Config.NUM_CATEGORIES, activation='softmax')
```

Εικόνα 6.20: Παράδειγμα επιπέδου εξόδου

Με το Sequential(model_layers), δημιουργείται το μοντέλο νευρωνικού δικτύου στοιβάζοντας τα στρώματα που ορίζονται στο model_layers το ένα μετά το άλλο σύμφωνα με τη σειρά στη λίστα. Το τελικό μοντέλο που υλοποιήθηκε, με όλα τα επίπεδα έχει την παρακάτω μορφή:

```python
from tensorflow import keras
from config import Config

def build_model():

    # Build the model
    model_layers = [
        keras.layers.Conv2D(filters=16, kernel_size=(3, 3), activation='relu',
                            input_shape=(Config.IMG_HEIGHT, Config.IMG_WIDTH, Config.CHANNELS)),
        keras.layers.BatchNormalization(),
        keras.layers.Conv2D(filters=32, kernel_size=(3, 3), activation='relu'),
        keras.layers.MaxPool2D(pool_size=(4, 4)),
        keras.layers.BatchNormalization(),

        keras.layers.Conv2D(filters=64, kernel_size=(3, 3), activation='relu'),
        keras.layers.Conv2D(filters=128, kernel_size=(3, 3), activation='relu'),
        keras.layers.MaxPool2D(pool_size=(4, 4)),
        keras.layers.BatchNormalization(),

        keras.layers.Conv2D(filters=256, kernel_size=(3, 3), activation='relu'),
        keras.layers.MaxPool2D(pool_size=(4, 4)),
        keras.layers.BatchNormalization(),

        keras.layers.Flatten(),
        keras.layers.Dense(512, activation='relu'),
        keras.layers.BatchNormalization(),
        keras.layers.Dropout(rate=0.5),

        keras.layers.Dense(Config.NUM_CATEGORIES, activation='softmax')
    ]
    model = keras.models.Sequential(model_layers)

    return model
```

Εικόνα 6.21: Τελικό μοντέλο

6.6.4 Επιστροφή τιμών

Τέλος, η συνάρτηση επιστρέφει το δημιουργημένο μοντέλο το οποίο και θα χρησιμοποιηθεί αργότερα στην εκπαίδευση.

```python
return model
```

Εικόνα 6.22: Επιστρεφόμενο μοντέλο

6.7 Εκπαίδευση μοντέλου

Το τελικό στάδιο περιλαμβάνει την εκπαίδευση του μοντέλου που κατασκευάσαμε. Η εκπαίδευση αναφέρεται στην επαναληπτική διαδικασία προσαρμογής των παραμέτρων του μοντέλου για την ελαχιστοποίηση μιας προκαθορισμένης συνάρτησης απωλειών. Αυτή η διαδικασία βελτιστοποίησης περιλαμβάνει το πέρασμα των παρτίδων δεδομένων εισόδου, τον υπολογισμό των προβλέψεων του μοντέλου, τη σύγκριση τους με τους πραγματικούς στόχους και την ενημέρωση των παραμέτρων με τη

χρήση τεχνικών όπως η διαβάθμιση κλίσης (μέθοδος βελτιστοποίησης της απόδοσης ενός νευρωνικού δικτύου, μειώνοντας το ποσοστό απώλειας/λάθους του δικτύου). Το σύνολο των δεδομένων επικύρωσης χρησιμοποιείται για την παρακολούθηση των επιδόσεων του μοντέλου σε αθέατα δεδομένα κατά τη διάρκεια της εκπαίδευσης, βοηθώντας στην αποφυγή της υπερπροσαρμογής και διασφαλίζοντας ότι το μοντέλο γενικεύεται καλά σε νέα παραδείγματα. Τελικά, ο στόχος της εκπαίδευσης είναι να βρεθεί το σύνολο των παραμέτρων που έχουν την καλύτερη απόδοση τόσο στα δεδομένα εκπαίδευσης, όσο και στα δεδομένα επικύρωσης, παράγοντας έτσι ένα μοντέλο που μπορεί να προβλέψει με ακρίβεια τα αποτελέσματα για αθέατες περιπτώσεις. Ο κώδικας που υλοποιήθηκε για την εκπαίδευση του μοντέλου μπορεί να χωριστεί σε τρία μέρη για καλύτερη κατανόηση. Την ρύθμιση του μοντέλου, την διαμόρφωση της εκπαίδευσης και τέλος την εκπαίδευση και αποθήκευση του μοντέλου.

6.7.1 Εισαγωγή βιβλιοθηκών

Για αρχή εισάγονται οι απαραίτητες βιβλιοθήκες που θα χρειαστούν κατά την εκπαίδευση. Η βιβλιοθήκη 'tensorflow.keras.callbacks', παρέχει μια σειρά από βοηθητικά προγράμματα για την προσαρμογή και τη βελτίωση της διαδικασίας εκπαίδευσης των νευρωνικών δικτύων. Εδώ χρησιμοποιείται το LearningRateScheduler όπου μας επιτρέπει τη δυναμική προσαρμογή του ρυθμού μάθησης κατά τη διάρκεια της εκπαίδευσης. Είναι ιδιαίτερα χρήσιμο για εργασίες όπου ο ιδανικός ρυθμός μάθησης μπορεί να αλλάξει καθώς προχωρά η βελτιστοποίηση, βοηθώντας στη λεπτομερή ρύθμιση της απόδοσης του μοντέλου. Το ModelCheckpoint, από την άλλη πλευρά, χρησιμοποιείται για την αποθήκευση του μοντέλου κατά τη διάρκεια της εκπαίδευσης. Μας επιτρέπει να παρακολουθούμε μια επιλεγμένη μετρική, στην περίπτωσή μας το val_accuracy, και να αποθηκεύουμε την κατάσταση του μοντέλου κάθε φορά που αυτή η μετρική βελτιώνεται. Η βιβλιοθήκη 'tensorflow.keras.optimizers' παρέχει μια συλλογή αλγορίθμων βελτιστοποίησης απαραίτητων για την αποτελεσματική εκπαίδευση νευρωνικών δικτύων. Εμείς χρησιμοποιούμε τον 'Adam' όπου ξεχωρίζει λόγω του learning rate μηχανισμού. Ο Adam συνδυάζει τα πλεονεκτήματα τόσο του adaptive learning rate, όπου προσαρμόζεται με βάση τα μεγέθη των κλίσεων, όσο και της ορμής (momentum), ο οποίος εξομαλύνει τη διαδρομή βελτιστοποίησης. Η βιβλιοθήκη 'tensorflow.keras.preprocessing.image' παρέχει βοηθητικά προγράμματα για την αποτελεσματική φόρτωση και προεπεξεργασία των δεδομένων εικόνας. Εμείς χρησιμοποιούμε το ImageDataGenerator, το οποίο είναι ένα ισχυρό εργαλείο για την επαύξηση δεδομένων σε πραγματικό χρόνο κατά τη διάρκεια της εκπαίδευσης του μοντέλου. Δημιουργεί παρτίδες επαυξημένων δεδομένων εικόνας, επιτρέποντας τις δυναμικές μεταβολές στα δεδομένα εισόδου (όπως περιστροφή, μετατόπιση, μεγέθυνση και αναστροφή) βοηθώντας στην διαφοροποίηση και τον εμπλουτισμό του συνόλου δεδομένων εκπαίδευσης. Αυτή η διαδικασία βοηθά στην βελτίωση της γενίκευσης του μοντέλου. Η βιβλιοθήκη 'math' της Python παρέχει ένα ευρύ φάσμα μαθηματικών συναρτήσεων. Εδώ χρησιμοποιείται η συνάρτηση math.floor(), όπου πραγματοποιεί στρογγυλοποίηση στον πλησιέστερο ακέραιο αριθμό. Τέλος, η Config είναι μια κλάση φτιαγμένη από εμάς που περιέχει διάφορες σταθερές και παραμέτρους που χρησιμοποιούνται σε όλο το έργο.

```
from tensorflow.keras.callbacks import LearningRateScheduler, ModelCheckpoint
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import math
from config import Config
```

Εικόνα 6.23: Απαραίτητες βιβλιοθήκες

6.7.2 Προκαθορισμένες τιμές

Σε αυτό το σημείο θα δούμε τις προκαθορισμένες τιμές που θα χρησιμοποιηθούν σε αυτό το μέρος.

Για αρχή, καθορίζονται οι τιμές για την σταθερά LEARNING_RATE και το dictionary LR_SCHEDULE_PARAMS, όπου θα χρησιμοποιηθούν για την καθορισμό της στρατηγικής learning rate scheduling. Όπως θα δούμε αναλυτικότερα παρακάτω, είναι μια στρατηγική όπου προσαρμόζεται δυναμικά ο ρυθμός μάθησης κατά την διάρκεια της εκπαίδευσης. Πιο συγκεκριμένα, με το LEARNING_RATE στο 0.001 ορίζεται ο αρχικός ρυθμός εκμάθησης για τον βελτιστοποιητή Adam όπου χρησιμοποιεί αυτόν τον ρυθμό εκμάθησης για να ενημερώσει τα βάρη του νευρωνικού δικτύου κατά τη διάρκεια της εκπαίδευσης. Ο ρυθμός εκμάθησης που επιλέξαμε δεν είναι ούτε πολύ υψηλός, ώστε να προκαλέσει ταλαντώσεις, ούτε πολύ χαμηλός ώστε να επιβραδύνει σημαντικά τη διαδικασία εκμάθησης. Στο dictionary LR_SCHEDULE_PARAMS δίνεται η τιμή στο initial_lr για να οριστεί ο αρχικός ρυθμός εκμάθησης για τον learning rate scheduler. Δίνουμε ίδια τιμή με το LEARNING_RATE, που σημαίνει ότι ξεκινάμε με τον ίδιο ρυθμό εκμάθησης με τον βελτιστοποιητή. Η παράμετρος drop καθορίζει τον παράγοντα με τον οποίο θα μειωθεί ο ρυθμός εκμάθησης. Στην περίπτωσή μας, ο ρυθμός εκμάθησης πολλαπλασιάζεται επι 0.5 (πτώση = 0.5) ανά κάποιες εποχές. Το ανά πόσες εποχές θα γίνεται αυτή η μείωση καθορίζεται από το epochs_drop. Έχει οριστεί σε 10, επομένως σε αυτήν την περίπτωση ο ρυθμός εκμάθησης θα μειώνεται κατά 0,5 κάθε 10 εποχές.

```
LEARNING_RATE = 0.001
LR_SCHEDULE_PARAMS = {
    "initial_lr": LEARNING_RATE,
    "drop": 0.5,
    "epochs_drop": 10
}
```

Εικόνα 6.24: Τιμές για το learning rate scheduling

Στην συνέχεια, ορίζονται οι τιμές στο dictionary AUGMENTATION_PARAMS. Αυτές οι τιμές θα χρησιμοποιηθούν για μια διαδικασία, η οποία ονομάζεται επαύξηση δεδομένων (data augmentation), κατά την οποία αυξάνεται τεχνητά το μέγεθος του συνόλου των δεδομένων εκπαίδευσης, εφαρμόζοντας διάφορες τροποποιήσεις στις υπάρχουσες εικόνες [40]. Παρόλο που δεν είναι απαραίτητο να χρησιμοποιηθεί αυτή η τεχνική, βοηθά στην έκθεση του μοντέλου σε μια ευρύτερη ποικιλία παραλλαγών στα δεδομένα εισόδου, η οποία με τη σειρά της μπορεί να οδηγήσει σε καλύτερη γενίκευση του εκπαιδευμένου μοντέλου. Πιο συγκεκριμένα, δίνεται στο rotation_range η τιμή 10 ώστε να περιστρέφονται οι τιμές συν-πλην 10 μοίρες, στο zoom_range η τιμή 0.15 για να ορίσουμε το εύρος

για την τυχαία μεγέθυνση ή σμίκρυνση των εικόνων, στο width_shift_range και height_shift_range η τιμή 0.1 ώστε να γίνονται τυχαίες μετατοπίσεις των εικόνων οριζόντια και κάθετα, στο shear_range η τιμή 0.15, όπου θα γίνεται μετατόπιση ενός τμήματος της εικόνας σε σχέση με ένα άλλο κατά μήκος του οριζόντιου ή του κατακόρυφου άξονα, στο horizontal_flip και vertical_flip δόθηκε η τιμή false ώστε να μην εφαρμοστεί κάποια αναστροφή και τέλος στο fill_mode η τιμή 'nearest', ώστε να συμπληρώνονται τα pixels που λείπουν στην τροποποιημένη εικόνα με την πλησιέστερη τιμή pixel από την αρχική εικόνα. Οι τιμές επιλέχτηκαν μετά από αρκετές δοκιμές, μιας και αν χρησιμοποιήσουμε μικρές τιμές χάνονται τα πλεονεκτήματα της επαύξησης δεδομένων, αλλά αν χρησιμοποιήσουμε πολύ μεγάλες τιμές αυξάνεται ο κίνδυνος υπερπροσαρμογής και της δημιουργίας υπερβολικά παραμορφωμένων εικόνων που δεν αντιπροσωπεύουν ρεαλιστικά παραδείγματα.

```
AUGMENTATION_PARAMS = {
    "rotation_range": 10,
    "zoom_range": 0.15,
    "width_shift_range": 0.1,
    "height_shift_range": 0.1,
    "shear_range": 0.15,
    "horizontal_flip": False,
    "vertical_flip": False,
    "fill_mode": "nearest"
}
```

Εικόνα 6.25: Τιμές για την επαύξηση δεδομένων

Ορίζεται, επίσης ο αριθμός 32 για την σταθερά BATCH_SIZE, όπου θα καθορίσει τον αριθμό των δειγμάτων που θα διαδοθούν μέσω του νευρωνικού δικτύου ταυτόχρονα κατά τη διάρκεια της εκπαίδευσης. Πιο συγκεκριμένα, στην εκπαίδευση νευρωνικών δικτύων, το σύνολο δεδομένων, συνήθως χωρίζεται σε μικρότερες παρτίδες για λόγους αποτελεσματικότητας. Κάθε παρτίδα τροφοδοτείται στο δίκτυο και οι κλίσεις υπολογίζονται με βάση τις απώλειες της συγκεκριμένης παρτίδας. Αυτές οι κλίσεις χρησιμοποιούνται στη συνέχεια για την ενημέρωση των βαρών του μοντέλου [41]. Επιλέξαμε έναν αριθμό που μετά από δοκιμές αποφασίσαμε ότι δίνει μια ισορροπία ανάμεσα στην αποτελεσματικότητα την εκπαίδευσης και στο χρόνο και τους πόρους που απαιτείται για αυτήν.

Ορίζεται στην συνέχεια η σταθερά EPOCHS σε 30. Στην μηχανική μάθηση, μια εποχή αντιπροσωπεύει ένα μοναδικό πέρασμα ολόκληρου του συνόλου δεδομένων μέσω του νευρωνικού δικτύου για την ενημέρωση των παραμέτρων του μοντέλου (βάρη και προκαταλήψεις). Η εκπαίδευση για πολλαπλές εποχές επιτρέπει στο μοντέλο να μαθαίνει από το σύνολο δεδομένων επαναληπτικά, βελτιώνοντας τις παραμέτρους του για να βελτιώσει την απόδοση. Ένας μικρός αριθμός εποχών μπορεί να οδηγήσει σε υπεραπλούστευση και μικρή γενίκευση ενώ ένας μεγάλος αριθμός σε αυξημένο χρόνο εκπαίδευσης και υπερπροσαρμογή [42]. Μετά από πειραματισμούς με διάφορες τιμές, καταλήξαμε στην παραπάνω τιμή ώστε να υπάρξει μια ισορροπία μεταξύ του χρόνου εκπαίδευσης και της απόδοσης του μοντέλου.

Τέλος, ορίστηκε η διαδρομή του υπολογιστή στην οποία θα αποθηκευτεί το εκπαιδευμένο μοντέλο καθώς και το όνομα του αρχείου του μοντέλου. Το αρχείο του μοντέλου είναι τύπου .h5 (Hierarchical Data Format version 5), όπου χρησιμοποιείται για αποθήκευση και οργάνωση μεγάλου όγκου δεδομένων.

```
BATCH_SIZE = 32
EPOCHS = 30
TRAINED_MODEL_FILENAME = "models/trained_model.h5"
```

Εικόνα 6.26: Υπόλοιπες σταθερές της κλάσης

6.7.3 Ρύθμιση μοντέλου

Ξεκινώντας από το πρώτο μέρος, με την ρύθμιση του μοντέλου, ορίζεται η συνάρτηση step_decay. Χρησιμοποιούμε αυτήν την συνάρτηση ώστε να ορίσει τον ρυθμό μάθησης κατά τη διάρκεια της εκπαίδευσης. Αυτό συμβαίνει επειδή επιλέξαμε την Learning Rate Scheduling προσέγγιση, αντί της Fixed learning rate όπου αντί ο ρυθμός μάθησης να είναι σταθερός κατά την εκπαίδευση, αυτός μεταβάλλεται. Η τεχνική αυτή βελτιστοποιεί τη σύγκλιση του μοντέλου, εξασφαλίζοντας ταχύτερη και σταθερότερη μάθηση [43]. Ο μαθηματικός υπολογισμός εντός της συνάρτησης συνδυάζει τις παραμέτρους που αναλύσαμε παραπάνω για τον υπολογισμό του ρυθμού μάθησης για κάθε εποχή. Για παράδειγμα, ας υποθέσουμε ότι η τρέχουσα εποχή είναι η 15. Σύμφωνα με τη διαμόρφωση, ο αρχικός ρυθμός μάθησης (initial_lr) είναι 0,001, ο παράγοντας εγκατάλειψης (drop) είναι 0,5 και ο αριθμός των εποχών προς εγκατάλειψη (epochs_drop) είναι 10. Αρχικά, υπολογίζουμε το (1 + epoch) / epochs_drop, το οποίο σε αυτή την περίπτωση είναι (1 + 15) / 10 = 1,6. Η συνάρτηση math.floor() στρογγυλοποιεί αυτή την τιμή στον πλησιέστερο ακέραιο, με αποτέλεσμα να προκύπτει το 1. Στη συνέχεια, αυξάνουμε τον συντελεστή πτώσης στη δύναμη της στρογγυλοποιημένης προς τα κάτω τιμής: 0,5^1 = 0,5. Τέλος, πολλαπλασιάζουμε τον αρχικό ρυθμό μάθησης με τον υπολογισμένο παράγοντα πτώσης για να λάβουμε τον ενημερωμένο ρυθμό μάθησης για την εποχή 15 που είναι 0,001 * 0,5 = 0,0005. Επομένως, για την εποχή 15, ο ρυθμός μάθησης θα μειωθεί σε 0,0005. Αυτή η διαδικασία συνεχίζεται για κάθε εποχή, μειώνοντας σταδιακά το ρυθμό μάθησης με βάση το καθορισμένο χρονοδιάγραμμα.

```
def step_decay(epoch):
    return Config.LR_SCHEDULE_PARAMS["initial_lr"] * pow(Config.LR_SCHEDULE_PARAMS["drop"],
        math.floor((1 + epoch) / Config.LR_SCHEDULE_PARAMS["epochs_drop"]))
```

Εικόνα 6.27: Η συνάρτηση step_decay

Συνεχίζοντας στην ρύθμιση του μοντέλου, δημιουργήθηκε η συνάρτηση train_model που δέχεται σαν όρισμα το μοντέλο και τα δεδομένα εικόνων. Είναι και αυτή η συνάρτηση που θα πραγματοποιήσει την εκπαίδευση του CNN. Η συνάρτηση αυτή περιλαμβάνει βασικά βήματα όπως η σύνταξη του μοντέλου, η διαμόρφωση της επαύξησης δεδομένων και η έναρξη της διαδικασίας εκπαίδευσης. Για αρχή επιλέγεται το μοντέλο εκπαίδευσης Adam με ρυθμό μάθησης 0,001 όπως είδαμε να έχει καθοριστεί στο LEARNING_RATE. Στην επομένη γραμμή, η μέθοδος compile καλείται στο αντικείμενο model για να διαμορφώσει το μοντέλο για εκπαίδευση. Πιο συγκεκριμένα, με το loss='categorical_crossentropy' ορίζεται η συνάρτηση απώλειας, με το optimizer=opt ορίζεται ο βελτιστοποιητής που επιλέξαμε προηγουμένως και με το metrics=['accuracy'] καθορίζεται η μετρική αξιολόγησης που θα παρακολουθείται κατά τη διάρκεια της εκπαίδευσης. Εδώ, το 'accuracy' χρησιμοποιείται για την παρακολούθηση της ακρίβειας ταξινόμησης του μοντέλου στα σύνολα δεδομένων εκπαίδευσης και

επικύρωσης. Τέλος, η κλάση ImageDataGenerator χρησιμοποιείται για τη δημιουργία επαυξημένων δεδομένων εκπαίδευσης με βάση τις παραμέτρους που είδαμε στο AUGMENTATION_PARAMS.

```
def train_model(model, X_train, y_train, X_val, y_val):
    # Compile the model
    opt = Adam(learning_rate=Config.LEARNING_RATE)
    model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])

    # Data augmentation and training
    aug = ImageDataGenerator(**Config.AUGMENTATION_PARAMS)
```

Εικόνα 6.28: Ρύθμιση μοντέλου

6.7.4 Διαμόρφωση της εκπαίδευσης

Συνεχίζοντας με το δεύτερο μέρος, την διαμόρφωση της εκπαίδευσης, χρησιμοποιώντας την κλάση LearningRateScheduler δημιουργείται ένα learning rate scheduler callback. Αυτή η κλήση επιτρέπει τη δυναμική προσαρμογή του ρυθμού μάθησης κατά τη διάρκεια της εκπαίδευσης με βάση προκαθορισμένους κανόνες. Εδώ, η step_decay περνάει ως η συνάρτηση που είναι υπεύθυνη για τον καθορισμό του ρυθμού μάθησης σε κάθε εποχή. Η παράμετρος verbose=1 ενεργοποιεί το verbose output, παρέχοντας πληροφορίες σχετικά με το learning rate schedule κατά τη διάρκεια της εκπαίδευσης.

```
# Create a learning rate scheduler callback
lr_scheduler = LearningRateScheduler(step_decay, verbose=1)
```

Εικόνα 6.29: Δημιουργία learning rate scheduler callback

Τέλος, δημιουργείται ένα model checkpoint callback με την χρήση της κλάσης ModelCheckpoint. Αυτό το callback είναι υπεύθυνο για την αποθήκευση των βαρών του μοντέλου κατά τη διάρκεια της εκπαίδευσης. Το Config.TRAINED_MODEL_FILENAME καθορίζει το όνομα αρχείου για την αποθήκευση του εκπαιδευμένου μοντέλου. Η παράμετρος monitor='val_accuracy' δίνει εντολή στο callback να παρακολουθεί τη μετρική της ακρίβειας επικύρωσης, διασφαλίζοντας ότι αποθηκεύεται το μοντέλο με την υψηλότερη ακρίβεια επικύρωσης. Εάν δεν υπήρχε αυτός ο μηχανισμός, θα αποθηκευόταν πάντα στο μοντέλο της τελευταίας εποχής, παρόλο που μπορεί να μην ήταν το βέλτιστο. Η παράμετρος save_best_only=True είναι αυτή που διασφαλίζει ότι αποθηκεύεται μόνο το καλύτερο μοντέλο, ενώ η παράμετρος mode='max' καθορίζει ότι μια υψηλότερη ακρίβεια επικύρωσης θεωρείται καλύτερη. Η παράμετρος verbose=1 ενεργοποιεί το verbose output, παρέχοντας πληροφορίες σχετικά με τη διαδικασία αποθήκευσης κατά τη διάρκεια της εκπαίδευσης. Μαζί, αυτές οι γραμμές δημιουργούν δύο βασικά callbacks που ενισχύουν τη διαδικασία εκπαίδευσης με τη δυναμική προσαρμογή του ρυθμού μάθησης και την αποθήκευση των checkpoints του μοντέλου με τις καλύτερες επιδόσεις για μεταγενέστερη χρήση.

```
# Model checkpoint callback
checkpoint = ModelCheckpoint(Config.TRAINED_MODEL_FILENAME, monitor='val_accuracy',
                             save_best_only=True, mode='max', verbose=1)
```

Εικόνα 6.30: Δημιουργία ένα model checkpoint callback

6.7.5 Εκπαίδευση και αποθήκευση μοντέλου

Τελειώνουμε με το τρίτο και τελευταίο μέρος, με την συνάρτηση fit που εκπαιδεύει το μοντέλο μας. Πιο συγκεκριμένα, με το aug.flow(X_train, y_train, batch_size=Config.BATCH_SIZE) δημιουργούνται επαυξημένες παρτίδες δεδομένων εκπαίδευσης χρησιμοποιώντας το μέγεθος παρτίδας (BATCH_SIZE) που έχει οριστεί. Η μέθοδος ροής ImageDataGenerator χρησιμοποιείται για την εφαρμογή τεχνικών επαύξησης δεδομένων στα δεδομένα εκπαίδευσης. Με το steps_per_epoch=len(X_train) // Config.BATCH_SIZE καθορίζεται ο αριθμός των παρτίδων που θα υποβάλλονται σε επεξεργασία σε κάθε εποχή, διαιρώντας τον συνολικό αριθμό των δειγμάτων εκπαίδευσης (len(X_train)) με το μέγεθος της παρτίδας. Με το epochs=Config.EPOCHS καθορίζει τον αριθμό των εποχών για τις οποίες θα πρέπει να εκπαιδευτεί το μοντέλο. Με το validation_data=(X_val, y_val) παρέχονται τα δεδομένα επικύρωσης για την αξιολόγηση της απόδοσης του μοντέλου μετά από κάθε εποχή όπου αποτελείται από τις εικόνες επικύρωσης (X_val) και τις αντίστοιχες ετικέτες τους (y_val). Τέλος, με το callbacks=[lr_scheduler, checkpoint] δίνονται τα callbacks που θα εκτελεσθούν κατά τη διάρκεια της εκπαίδευσης για να προσαρμόσουν το ρυθμό μάθησης και να αποθηκεύσουν τα σημεία ελέγχου του μοντέλου με τις καλύτερες επιδόσεις.

```
# Train the model with the learning rate scheduler callback
history = model.fit(
    aug.flow(X_train, y_train, batch_size=Config.BATCH_SIZE),
    steps_per_epoch=len(X_train) // Config.BATCH_SIZE,
    epochs=Config.EPOCHS,
    validation_data=(X_val, y_val),
    callbacks=[lr_scheduler, checkpoint]
)
```

Εικόνα 6.31: Εκπαίδευση μοντέλου

Μετά την ολοκλήρωση της εκπαίδευσης, με την γραμμή model.save(Config.TRAINED_MODEL_FILENAME) θα αποθηκευτεί το εκπαιδευμένο μοντέλο στην καθορισμένη διαδρομή. Το εκπαιδευμένο μοντέλο περιλαμβάνει τα μαθημένα βάρη και την αρχιτεκτονική, επιτρέποντας την επαναχρησιμοποίησή του για εξαγωγή συμπερασμάτων ή περαιτέρω εκπαίδευση στο μέλλον.

```
# Save the trained model
model.save(Config.TRAINED_MODEL_FILENAME)
```

Εικόνα 6.32: Αποθήκευση μοντέλου

Τέλος, επιστρέφονται από τη συνάρτηση το εκπαιδευμένο μοντέλο (model) και το ιστορικό εκπαίδευσης (history). Το ιστορικό εκπαίδευσης περιέχει πληροφορίες σχετικά με τις μετρικές εκπαίδευσης και επικύρωσης (π.χ. απώλεια και ακρίβεια) που καταγράφηκαν κατά τη διάρκεια της διαδικασίας εκπαίδευσης, οι οποίες μπορούν να χρησιμοποιηθούν για ανάλυση.

```
# Return the trained model
return model, history
```

Εικόνα 6.33: Επιστροφή μοντέλου

6.8 Επίλογος

Συμπερασματικά λοιπόν, σε αυτό το κεφάλαιο ορίστηκαν αρκετές βασικές έννοιες που αφορούν την μηχανική μάθηση, μεταξύ των οποίων CNN, Keras, Tensorflow, εκπαίδευση, υπεραπλούστευση και υπερπροσαρμογή. Στην συνέχεια, είδαμε την δομή και την υλοποίηση του κώδικα, από το πρώτο βήμα που είναι η φόρτωση και η προεπεξεργασία των δεδομένων εκπαίδευσης, το επόμενο που αφορά την κατασκευή του μοντέλου μέχρι το τελευταίο που είναι η εκπαίδευση του μοντέλου.

Κεφάλαιο 7ο: Πρόβλεψη χρησιμοποιώντας το εκπαιδευόμενο μοντέλο

7.1 Εισαγωγή

Προκειμένου να δοκιμάσουμε το μοντέλο που εκπαιδεύσαμε, έχει δημιουργηθεί μια εφαρμογή με γραφικό περιβάλλον χρήστη (GUI). Σε αυτήν την εφαρμογή ο χρήστης μπορεί να φορτώσει την εικόνα που επιθυμεί, και πατώντας ένα κουμπί να του εμφανιστεί η πρόβλεψη του μοντέλου σχετικά με την ύπαρξη ή όχι κάποιου deepfake σε αυτήν την εικόνα.

Χρησιμοποιεί την βιβλιοθήκη PyQt5 για το γραφικό περιβάλλον και το TensorFlow/Keras για την φόρτωση του μοντέλου και την πρόβλεψη. Το PyQt5 είναι μια βιβλιοθήκη της Python που μας επιτρέπει να δημιουργούμε εφαρμογές με γραφικό περιβάλλον. Παρέχει ένα ευρύ φάσμα στοιχείων GUI, όπως κουμπιά, ετικέτες, πεδία κειμένου, παράθυρα διαλόγου και διατάξεις [44].

7.2 Εισαγωγή βιβλιοθηκών

Για αρχή εισάγονται οι απαραίτητες βιβλιοθήκες. Πιο συγκεκριμένα εισάγεται η 'sys' για την αλληλεπίδραση με το περιβάλλον εκτέλεσης της Python. Στην συνέχεια, εισάγονται οι PyQt5 βιβλιοθήκες για τη δημιουργία γραφικού περιβάλλοντος. Πιο συγκεκριμένα, από το 'PyQt5.QtWidgets' εισάγεται το 'QApplication' για να γίνει διαχείριση της ροή και τις κύριες ρυθμίσεις της GUI εφαρμογής, το 'QWidget' όπου είναι η βασική κλάση για όλα τα UI αντικείμενα παρέχοντας βασικές δυνατότητες για την δημιουργία παραθύρων, το 'QLabel' όπου χρησιμοποιείται για την εμφάνιση κειμένου και εικόνων, το "QVBoxLayout, που πρόκειται για έναν διαχειριστή διάταξης που τακτοποιεί τα widgets κάθετα, το 'QPushButton' για να εισαχθούν κουμπιά με την δυνατότητα να πατιούνται , το 'QFileDialog' όπου παρέχει ένα παράθυρο διαλόγου που επιτρέπει στους χρήστες να επιλέγουν αρχεία ή καταλόγους και τέλος το 'QHBoxLayout', όπου είναι ένας διαχειριστής διάταξης που τακτοποιεί τα widgets οριζόντια. Στην συνέχεια, εισάγεται η 'load_model' απο 'tensorflow.keras' για το φόρτωμα του μοντέλου, η 'np' από 'numpy' για τον χειρισμό συστοιχιών και μαθηματικές πράξεις, η 'cv2' για τη φόρτωση και την αλλαγή μεγέθους των εικόνων και τέλος η Config που περιέχει διάφορες σταθερές και παραμέτρους που χρησιμοποιούνται σε όλο το έργο.

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QLabel, QVBoxLayout, QPushButton, QFileDialog, QHBoxLayout
from PyQt5.QtGui import QPixmap
from PyQt5.QtCore import Qt
from tensorflow.keras.models import load_model
import numpy as np
import cv2
from config import Config
```

Εικόνα 7.1: Απαραίτητες βιβλιοθήκες

7.3 Προκαθορισμένες τιμές

Σε αυτό το σημείο θα δούμε τις προκαθορισμένες τιμές που θα χρησιμοποιηθούν σε αυτό το μέρος. Ορίζεται το λεξικό CLASSES ώστε να δείξει την απαραίτητη πρόβλεψη με βάση το αποτέλεσμα του μοντέλου. Στην συνέχεια, ορίζεται το ύψος και το πλάτος της εικόνας σε pixels, επομένως καθορίζεται ο τελικός αριθμός των pixels που θα έχει η κάθε εικόνα. Επιλέξαμε τις διαστάσεις 128 για το ύψος και 128 για το πλάτος και τις αποθηκεύσαμε στις σταθερές IMG_HEIGHT και IMG_WIDTH. Τέλος, ορίστηκε η διαδρομή του υπολογιστή στην οποία θα βρει το εκπαιδευμένο μοντέλο με βάση το όνομα του. Το αρχείο του μοντέλου είναι τύπου .h5 (Hierarchical Data Format version 5), όπου χρησιμοποιείται για αποθήκευση και οργάνωση μεγάλου όγκου δεδομένων.

```
CLASSES = {0: 'Original image', 1: 'Deepfake image'}
IMG_HEIGHT, IMG_WIDTH = 128, 128
TRAINED_MODEL_FILENAME = "models/trained_model.h5"
```

Εικόνα 7.2: Σταθερές της κλάσης

7.4 Διεπαφή χρήστη

Η κλάση ImagePredictorUI χρησιμεύει ως διεπαφή χρήστη (UI) για την εφαρμογή που υλοποιούμε. Κληρονομεί από την QWidget, η οποία είναι η βασική κλάση για όλα τα αντικείμενα διεπαφής χρήστη στην PyQt5, επιτρέποντας την να αντιμετωπίζεται ως γραφικό στοιχείο (widget). Παρακάτω θα παρουσιάσουμε την υλοποίησή της χωρίζοντας την σε πέντε κατηγορίες. Την ρύθμιση του παραθύρου, την δημιουργία των γραφικών στοιχείων, την ρύθμιση της διάταξης, την σύνδεση των κουμπιών με λειτουργίες και τέλος την μορφοποίηση.

7.4.1 Ρύθμιση παραθύρου

Κατά την ρύθμιση του παραθύρου αρχικά με το setWindowTitle ορίζεται ο τίτλος που θα έχει το παράθυρο. Για τίτλο έχουμε επιλέξει το 'Deepfake Detector '.

```
self.setWindowTitle('Deepfake Detector')
```

Εικόνα 7.3: Ορισμός τίτλου παραθύρου

Στην συνέχεια δημιουργείται το widget QLabel με όνομα image_label και προστίθεται το κύριο παράθυρο (self). Αυτή η ετικέτα θα χρησιμοποιηθεί για την εμφάνιση της εικόνας. Με το setAlignment(Qt.AlignCenter) ορίζεται ότι το περιεχόμενο της ετικέτας θα κεντράρεται οριζόντια και με το setFixedSize(400, 400) ορίζεται το σταθερό μέγεθος της ετικέτας σε 400x400 pixels.

```
self.image_label = QLabel(self)
self.image_label.setAlignment(Qt.AlignCenter)
self.image_label.setFixedSize(400, 400)
```

Εικόνα 7.4: Ετικέτα για εμφάνιση εικόνας

Παρομοίως, δημιουργείται ακόμα ένα widget QLabel αλλά με όνομα result_label και προστίθεται στο κύριο παράθυρο. Αυτή η ετικέτα θα εμφανίζει το αποτέλεσμα της πρόβλεψης. Με το setAlignment(Qt.AlignCenter) ορίζεται ότι το περιεχόμενο της ετικέτας θα κεντράρεται οριζόντια και τέλος με το setObjectName('resultLabel') θέτεται το όνομα του αντικειμένου για λόγους μορφοποίησης, επιτρέποντας το να χρησιμοποιηθεί αργότερα κατά την εφαρμογή ενός style sheet.

```
self.result_label = QLabel(self)
self.result_label.setAlignment(Qt.AlignCenter)
self.result_label.setObjectName('resultLabel')
```

Εικόνα 7.5: Ετικέτα για αποτέλεσμα πρόβλεψης

Αυτές οι γραμμές είναι ζωτικής σημασίας για τον καθορισμό της βασικής δομής του παραθύρου, συμπεριλαμβανομένου του τίτλου του και των ετικετών για την εμφάνιση των εικόνων και των αποτελεσμάτων της πρόβλεψης.

7.4.2 Δημιουργία των γραφικών στοιχείων

Σε αυτό το σημείο δημιουργείται ένα widget QPushButton με όνομα upload_button με το κείμενο "Upload Image", και προστίθεται στο κύριο παράθυρο. Αυτό το κουμπί θα χρησιμοποιηθεί ώστε ο χρήστης να ανεβάσει την επιθυμητή εικόνα. Με το setObjectName('uploadButton') ορίζεται το όνομα του αντικειμένου για λόγους μορφοποίησης, επιτρέποντας το να χρησιμοποιηθεί αργότερα κατά την εφαρμογή ενός style sheet.

```
self.upload_button = QPushButton('Upload Image', self)
self.upload_button.setObjectName('uploadButton')
```

Εικόνα 7.6: Κουμπί για ανέβασμα εικόνας

Ομοίως, δημιουργείται ακόμα ένα widget QPushButton με όνομα upload_button αλλά με το κείμενο "Make Prediction", και προστίθεται στο κύριο παράθυρο. Αυτό το κουμπί θα χρησιμοποιηθεί ώστε ο χρήστης να δει την τελική πρόβλεψη του μοντέλου. Με το setObjectName('predictButton') ορίζεται το όνομα του αντικειμένου για λόγους μορφοποίησης, επιτρέποντας το να χρησιμοποιηθεί αργότερα κατά την εφαρμογή ενός style sheet.

```
self.predict_button = QPushButton('Make Prediction', self)
self.predict_button.setObjectName('predictButton')
```

Εικόνα 7.7: Κουμπί για δημιουργία πρόβλεψης

Η δημιουργία widget είναι ζωτικής σημασίας για την παροχή διαδραστικών στοιχείων στη διεπαφή χρήστη. Αυτές οι γραμμές δημιουργούν κουμπιά για τη μεταφόρτωση εικόνων και την πραγματοποίηση προβλέψεων, βασικές λειτουργίες για την εφαρμογή deepfake detector.

7.4.3 Ρύθμιση διάταξης

Εδώ δημιουργείται ένα QVBoxLayout με το όνομα layout. Αυτή η διάταξη τοποθετεί τα widgets κάθετα. Η μέθοδος addWidget χρησιμοποιείται στη συνέχεια για την προσθήκη widgets σε αυτή τη διάταξη. Όπως φαίνεται στην παραπάνω εικόνα, το upload_button, το image_label, το predict_button και το result_label που έχουμε ήδη αναλύσει, προστίθενται στη διάταξη ένα προς ένα.

```
# Create a vertically layout for buttons
layout = QVBoxLayout()
layout.addWidget(self.upload_button)
layout.addWidget(self.image_label)
layout.addWidget(self.predict_button)
layout.addWidget(self.result_label)
```

Εικόνα 7.8: Κάθετη διάταξη widgets

Ομοίως, στην συνέχεια δημιουργείται μια διάταξη QHBoxLayout με το όνομα button_layout. Αυτή η διάταξη τοποθετεί τα widgets οριζόντια. Η μέθοδος addWidget χρησιμοποιείται στη συνέχεια για την προσθήκη των upload_button και predict_button που έχουμε ήδη αναλύσει, σε αυτή την οριζόντια διάταξη.

```
# Create a horizontal layout for buttons
button_layout = QHBoxLayout()
button_layout.addWidget(self.upload_button)
button_layout.addWidget(self.predict_button)
```

Εικόνα 7.9: Οριζόντια διάταξη widgets

Σε αυτό το σημείο η διάταξη button_layout (οριζόντια διάταξη) προστίθεται στην κύρια διάταξη (κατακόρυφη διάταξη) χρησιμοποιώντας τη μέθοδο addLayout. Αυτό εξασφαλίζει ότι τα κουμπιά upload και predict που έχουμε δει, τοποθετούνται οριζόντια μέσα στην κύρια κάθετη διάταξη.

```
# Add the button layout to the main layout
layout.addLayout(button_layout)
```

Εικόνα 7.10: Εισαγωγή οριζόντιας διάταξης στην κύρια

Τέλος, ορίζονται τα περιθώρια για την κύρια διάταξη. Τα περιθώρια προσθέτουν χώρο γύρω από το εξωτερικό της διάταξης. Εδώ, ορίζονται περιθώρια 20 pixels σε όλες τις πλευρές (αριστερά, πάνω, δεξιά, κάτω).

```
# Add some margins to the layout
layout.setContentsMargins(20, 20, 20, 20)
```

Εικόνα 7.11: Εισαγωγή περιθωρίων

Η ρύθμιση διάταξης είναι απαραίτητη για την οργάνωση και την τοποθέτηση των widgets στο κύριο παράθυρο. Οι διατάξεις QVBoxLayout και QHBoxLayout είναι ευρέως χρησιμοποιούμενες διατάξεις στο Qt για την κάθετη και οριζόντια διάταξη των widgets, αντίστοιχα.

7.4.4 Σύνδεση κουμπιών με λειτουργίες

Η πρώτη γραμμή κώδικα συνδέει το πάτημα (κλικ) του κουμπιού upload_button με τη συνάρτηση upload_image. Έτσι, όταν το κουμπί upload πατηθεί, θα εκτελεστεί η συνάρτηση upload_image όπου θα αναλυθεί παρακάτω.

Ομοίως, η επόμενη γραμμή συνδέει το σήμα πάτημα του κουμπιού predict_button με τη συνάρτηση make_prediction. Έτσι, όταν γίνεται κλικ στο κουμπί predict, θα εκτελεστεί η συνάρτηση make_prediction, όπου θα αναλυθεί παρακάτω.

```
# Connect button signals to functions
self.upload_button.clicked.connect(self.upload_image)
self.predict_button.clicked.connect(self.make_prediction)
```

Εικόνα 7.12: Σύνδεση κουμπιών με συναρτήσεις

Τα σήματα κουμπιών είναι απαραίτητα για τη σύνδεση των αλληλεπιδράσεων του χρήστη (στην προκειμένη περίπτωση, τα κλικ των κουμπιών) με συγκεκριμένες ενέργειες ή λειτουργίες στην εφαρμογή. Συνδέοντας τα κλικ των κουμπιών με τις αντίστοιχες λειτουργίες, επιτρέπεται στα κουμπιά να ενεργοποιούν την προβλεπόμενη λειτουργικότητα όταν αλληλεπιδρά ο χρήστης.

7.4.5 Μορφοποίηση

Τέλος, το style sheet με όνομα 'styles.qss' ανοίγει σε κατάσταση ανάγνωσης χρησιμοποιώντας τη συνάρτηση open. Έτσι, διαβάζονται τα περιεχόμενα του αρχείου με τη μέθοδο read και εφαρμόζονται στο κύριο παράθυρο (self) με τη μέθοδο setStyleSheet.

```
# Apply style sheet
with open('styles.qss', 'r') as style_file:
    self.setStyleSheet(style_file.read())
```

Εικόνα 7.13: Εφαρμογή του style sheet

Η διαμόρφωση είναι σημαντική για την προσαρμογή της εμφάνισης των στοιχείων της διεπαφής χρήστη. Όπως θα δούμε και παρακάτω, με την εφαρμογή ενός style sheet, δίνεται η δυνατότητα να τροποποιηθούν ιδιότητες όπως χρώματα, γραμματοσειρές, μεγέθη και αποστάσεις για να επιτευχθεί ο επιθυμητός οπτικός σχεδιασμός της εφαρμογής. Η χρήση εξωτερικών style sheets επιτρέπει την εύκολη τροποποίηση και συντήρηση των ιδιοτήτων διαμόρφωσης ξεχωριστά από τον κύριο κώδικα.

7.4.6 Το style sheet

Τα .qss, ή Qt Style Sheets, χρησιμεύουν ως ένας μηχανισμός μέσα στο πλαίσιο Qt για την προσαρμογή της οπτικής εμφάνισης των widgets στις εφαρμογές. Παρόμοια με τα CSS για την ανάπτυξη ιστοσελίδων, τα .qss επιτρέπουν στους προγραμματιστές να καθορίζουν στυλ όπως χρώματα, γραμματοσειρές και περιθώρια για διάφορα στοιχεία του UI, προσφέροντας μια προσαρμοσμένη και λειτουργική εμπειρία χρήστη σε όλες τις πλατφόρμες [45]. Ο διαχωρισμός του style sheet από τον κώδικα της εφαρμογής διευκολύνει την ευκολία συντήρησης και τις δυναμικές επιλογές styling, ενώ η ευελιξία και η ενσωμάτωσή του με τις εφαρμογές Qt ενισχύουν την παραγωγικότητα, επιτρέποντας αποδοτικές διαδικασίες σχεδιασμού και ανάπτυξης UI.

Στην αρχή, ορίζεται η επιθυμητή μορφοποίηση για το κουμπί Upload Button. Με το 'background-color' ορίζεται το χρώμα φόντου σε μια απόχρωση του πράσινου, με το 'color' ορίζεται το χρώμα κειμένου σε λευκό, με το 'border' ορίζεται ένα συμπαγές περίγραμμα 1px με το ίδιο πράσινο χρώμα, με το 'border-radius' δίνονται στο κουμπί στρογγυλεμένες γωνίες με ακτίνα 5px και τέλος προστίθεται 10px κενός χώρος ανάμεσα στο περίγραμμα και στο περιεχόμενο του κουμπιού.

```
/* Upload Button */
QPushButton#uploadButton {
    background-color: #4CAF50;
    color: white;
    border: 1px solid #4CAF50;
    border-radius: 5px;
    padding: 10px;
}
```

Εικόνα 7.14: Μορφοποίηση στο κουμπί upload

Για το ίδιο κουμπί ορίζεται το χρώμα που θα παίρνει όταν ο χρήστης μετακινεί το ποντίκι του πάνω σε αυτό (hover). Επιλέξαμε ένα πιο σκούρο πράσινο χρώμα ώστε ο χρήστης να παίρνει μια επιβεβαίωση ότι βρίσκεται πάνω στο κουμπί.

```
QPushButton#uploadButton:hover {
    background-color: #45a049;
}
```

Εικόνα 7.15: Αλλαγή χρώματος στο κουμπί upload

Στην συνέχεια, ορίζονται ακριβώς τα ίδια χαρακτηριστικά για το Predict Button. Με το 'background-color' ορίζεται το χρώμα φόντου σε μια απόχρωση του μπλέ, με το 'color' ορίζεται το χρώμα κειμένου σε λευκό, με το 'border' ορίζεται ένα συμπαγές περίγραμμα 1px με το ίδιο μπλε χρώμα, με το 'border-radius' δίνονται στο κουμπί στρογγυλεμένες γωνίες με ακτίνα 5px και τέλος προστίθεται 10px κενός χώρος ανάμεσα στο περίγραμμα και στο περιεχόμενο του κουμπιού.

```
/* Predict Button */
QPushButton#predictButton {
    background-color: #008CBA;
    color: white;
    border: 1px solid #008CBA;
    border-radius: 5px;
    padding: 10px;
}
```

Εικόνα 7.16: Μορφοποίηση στο κουμπί predict

Για το ίδιο κουμπί, ορίζεται το χρώμα που θα παίρνει όταν ο χρήστης μετακινεί το ποντίκι του πάνω σε αυτό (hover). Επιλέξαμε ένα πιο σκούρο μπλε χρώμα ώστε ο χρήστης να παίρνει μια επιβεβαίωση ότι βρίσκεται πάνω στο κουμπί.

```
QPushButton#predictButton:hover {
    background-color: #0071a0;
}
```

Εικόνα 7.17: Αλλαγή χρώματος στο κουμπί predict

Έπειτα, ορίζεται η επιθυμητή μορφοποίηση για την ετικέτα που εμφανίζει το αποτέλεσμα της πρόβλεψης. Με το 'color' ορίζεται το χρώμα κειμένου σε μαύρο ώστε να υπάρχει αντίθεση, ορίζεται το μέγεθος του κειμένου σε 18px και τέλος το κείμενο γίνεται bold ώστε να είναι πιο έντονο.

```
/* Result Label */
QLabel#resultLabel {
    color: #000000;
    font-size: 18px;
    font-weight: bold;
}
```

Εικόνα 7.18: Μορφοποίηση result label

Τέλος, ορίζεται το χρώμα φόντου χρησιμοποιώντας μια γραμμική διαβάθμιση από ένα ανοιχτό γκρι στο πάνω αριστερό μέρος έως ένα λευκό στο κάτω δεξί μέρος.

```
/* Main Window Background */
QWidget#mainWindow {
    background: qlineargradient(x1:0, y1:0, x2:1, y2:1, stop:0 #f0f0f0, stop:1 #ffffff);
}
```

Εικόνα 7.19: Μορφοποίηση κύριου παραθύρου

7.5 Υλοποίηση μεθόδων

Στην συνέχεια, δημιουργείται η κλάση ImagePredictorApp που κληρονομεί από την ImagePredictorUI και την επεκτείνει προσθέτοντας λειτουργικότητα ειδικά για την ανίχνευση deepfake. Κατά την αρχικοποίηση φορτώνει το εκπαιδευμένο μοντέλο και υλοποιεί μεθόδους για την φόρτωση της εικόνας, την προ εξεργασία της και την εμφάνιση της πρόβλεψης. Οι μέθοδοι που θα αναλύσουμε παρακάτω αφορούν το ανέβασμα της εικόνας, την φόρτωση και την αλλαγή του μεγέθους της, την προετοιμασία της εικόνας και τέλος την δημιουργία πρόβλεψης.

7.5.1 Κύρια μέθοδος

Ξεκινάμε με το κύριο μπλοκ όπου είναι το σημείο εκκίνησης στον κώδικα μας. Σε αυτό το σημείο αρχικοποιείται η εφαρμογή PyQt5 και δημιουργείται ενα instance της κλάσης ImagePredictorApp περνώντας την παράμετρο model_filename που ορίστηκε, όπως είδαμε κατά την αρχικοποιήση των σταθερών, ώστε να βρεί το αρχείο που περιέχει το μοντέλο. Στην συνέχεια, με την μέθοδο show() γίνεται ορατό το παράθυρο της εφαρμογής και τέλος, καλώντας την app.exec_(), ξεκινάει ο βρόγχος συμβάντων. Αυτή η μέθοδος περιμένει να συμβούν συμβάντα εντός της εφαρμογής και τερματίζει την εφαρμογή όταν κλείσει το κύριο παράθυρο. Η συνάρτηση sys.exit() διασφαλίζει ότι η εφαρμογή τερματίζεται σωστά.

Έτσι, ο κύριος σκοπός σε αυτό το σημείο είναι να αρχικοποιηθεί και να εκτελεστεί η εφαρμογή ImagePredictorApp.

```
if __name__ == '__main__':
    app = QApplication(sys.argv)
    predictor_app = ImagePredictorApp(model_filename=Config.TRAINED_MODEL_FILENAME)
    predictor_app.show()
    sys.exit(app.exec_())
```

Εικόνα 7.20: Κύρια μέθοδος

7.5.2 Ανέβασμα εικόνας

Η μέθοδος upload_image καλείται όταν πατηθεί το κουμπί 'Upload Image'. Επιτρέπει στους χρήστες να επιλέξουν ένα αρχείο εικόνας από το σύστημα τους και στην συνέχεια κάνει προεπισκόπηση της επιλεγμένης εικόνας στο UI και διατηρεί την διαδρομή της επιλεγμένης εικόνας για μεταγενέστερες λειτουργίες.

Πιο συγκεκριμένα, στην αρχή δημιουργεί ένα παράθυρο διαλόγου αρχείου (file_dialog) χρησιμοποιώντας το QFileDialog όταν πατηθεί το κουμπί 'Upload Image'. Έπειτα, ζητά από το χρήστη να επιλέξει ένα αρχείο εικόνας καλώντας τη μέθοδο getOpenFileName. Ο χρήστης μπορεί να επιλέξει αρχεία εικόνας με επεκτάσεις .jpg ή .png. Εάν ο χρήστης επιλέξει ένα αρχείο εικόνας, δηλαδή αν το image_path δεν είναι κενό, φορτώνει την εικόνα από την επιλεγμένη διαδρομή χρησιμοποιώντας τη μέθοδο load_and_resize_image που θα αναλυθεί παρακάτω. Στην συνέχεια, θέτει το φορτωμένο pixmap της εικόνας στο image_label, καθιστώντας το ορατό στο παράθυρο της εφαρμογής. Τέλος, αποθηκεύει την επιλεγμένη διαδρομή εικόνας στο χαρακτηριστικό self.image_path για μετέπειτα χρήση στην πρόβλεψη.

Συνολικά, αυτή η μέθοδος διευκολύνει την επιλογή και την εμφάνιση μιας εικόνας μέσα στο περιβάλλον εργασίας της εφαρμογής, όταν γίνεται κλικ στο κουμπί 'Upload Image'.

```
def upload_image(self):
    file_dialog = QFileDialog()
    image_path, _ = file_dialog.getOpenFileName(self, 'Select an image', '', 'Image files (*.jpg *.png)')

    if image_path:
        pixmap = self.load_and_resize_image(image_path)
        self.image_label.setPixmap(pixmap)
        self.image_label.show()
        self.image_path = image_path
```

Εικόνα 7.21: Μέθοδος για ανέβασμα εικόνας

7.5.3 Φόρτωση και αλλαγή μεγέθους εικόνας

Η μέθοδος load_and_resize_image καλείται μέσα από την μέθοδο upload_image και διασφαλίζει ότι η εικόνα που φορτώνεται έχει το κατάλληλο μέγεθος ώστε να χωράει στην καθορισμένη περιοχή του UI, διατηρώντας την αρχική αναλογία διαστάσεων για βέλτιστη ποιότητα εμφάνισης.

Πιο συγκεκριμένα, στην αρχή φορτώνει την εικόνα με βάση το image_path χρησιμοποιώντας το QPixmap, το οποίο παρέχει υποστήριξη για την εμφάνιση εικόνων σε εφαρμογές PyQt. Στην συνέχεια, αλλάζει το μέγεθος της φορτωμένης εικόνας (pixmap) ώστε να ταιριάζει στο μέγεθος του widget

image_label διατηρώντας την αναλογία διαστάσεων. Αυτό επιτυγχάνεται με τη χρήση της μεθόδου scaled της QPixmap, με το όρισμα Qt.KeepAspectRatio. Τέλος, επιστρέφει το αναδιαμορφωμένο σε μέγεθος pixmap, το οποίο μπορεί στη συνέχεια να οριστεί ως pixmap του widget image_label για την εμφάνιση της εικόνας στην εφαρμογή.

```
def load_and_resize_image(self, image_path):
    pixmap = QPixmap(image_path)
    scaled_pixmap = pixmap.scaled(self.image_label.size(), Qt.KeepAspectRatio)
    return scaled_pixmap
```

Εικόνα 7.22: Μέθοδος για φόρτωση και αλλαγή μεγέθους εικόνας

7.5.4 Δημιουργία πρόβλεψης

Η μέθοδος make_prediction καλείται όταν πατηθεί το κουμπί ‘Make Prediction’. Επιτρέπει στους χρήστες να λαμβάνουν προβλέψεις για εικόνες που έχουν μεταφορτωθεί και να προβάλλουν τα αποτελέσματα απευθείας μέσα στο περιβάλλον της εφαρμογής. Η μέθοδος προ-επεξεργάζεται την εικόνα, επεκτείνει τις διαστάσεις της ώστε να ταιριάζει με το αναμενόμενο σχήμα εισόδου του φορτωμένου μοντέλου και εκτελεί την πρόβλεψη χρησιμοποιώντας το μοντέλο. Ο δείκτης της προβλεπόμενης κλάσης προσδιορίζεται με την εύρεση του δείκτη της μέγιστης προβλεπόμενης τιμής από την έξοδο του μοντέλου. Στη συνέχεια, η αντίστοιχη ετικέτα για την προβλεπόμενη κλάση ανακτάται από το λεξικό Config.CLASSES. Τέλος, εμφανίζεται το αποτέλεσμα της πρόβλεψης.

Πιο συγκεκριμένα, στην αρχή ελέγχεται με το hasattr() αν το image_path υπάρχει για να διασφαλιστεί ότι μια εικόνα έχει μεταφορτωθεί πριν προσπαθήσει να κάνει μια πρόβλεψη. Εάν δεν έχει επιλεγεί καμία εικόνα, εμφανίζει ένα μήνυμα που υποδεικνύει ότι δεν έχει επιλεγεί καμία εικόνα (No image selected). Εάν υπάρχει διαθέσιμη εικόνα, την προ-επεξεργάζεται καλώντας τη μέθοδο preprocess_image, η οποία φορτώνει την εικόνα, αλλάζει το μέγεθός της και κανονικοποιεί τις τιμές των pixels. Έπειτα, μεταβιβάζει την προεπεξεργασμένη εικόνα στο φορτωμένο μοντέλο μάθησης (self.loaded_model) για πρόβλεψη όπου γίνεται με την κλήση της μεθόδου predict του μοντέλου. Για να γίνει η πρόβλεψη θα χρειαστεί να προσδιορίσει την προβλεπόμενη κλάση με την εύρεση του δείκτη της μέγιστης τιμής στον πίνακα πρόβλεψης. Αυτός ο δείκτης αντιστοιχεί στην προβλεπόμενη ετικέτα κλάσης. Στην συνέχεια, αναζητά την προβλεπόμενη ετικέτα κλάσης από το λεξικό Config.CLASSES που είδαμε, χρησιμοποιώντας τον δείκτη της προβλεπόμενης κλάσης. Τέλος, ορίζει το κείμενο του widget result_label για να εμφανίζει το αποτέλεσμα της πρόβλεψης, υποδεικνύοντας την ετικέτα της προβλεπόμενης κλάσης. Το αποτέλεσμα θα είναι 'Original image' ή 'Deepfake image'.

```
def make_prediction(self):
    if hasattr(self, 'image_path'):
        preprocessed_img = self.preprocess_image(self.image_path)
        preprocessed_img = np.expand_dims(preprocessed_img, axis=0)

        predicted_class = np.argmax(self.loaded_model.predict(preprocessed_img))
        predicted_label = Config.CLASSES.get(predicted_class, 'Unknown')

        self.result_label.setText('Prediction: ' + str(predicted_label))
    else:
        self.result_label.setText('No image selected')
```

Εικόνα 7.23: Μέθοδος για πρόβλεψη

7.5.5 Προετοιμασία εικόνας

Η μέθοδος preprocess_image έχει κληθεί ήδη πριν την δημιουργία πρόβλεψης, μέσα από την μέθοδο make_prediction και προετοιμάζει τη μεταφορτωμένη εικόνα για πρόβλεψη εκτελώντας τα απαραίτητα βήματα προεπεξεργασίας, όπως αλλαγή μεγέθους και κανονικοποίηση των pixels. Όταν η εικόνα μορφοποιείται κατάλληλα σύμφωνα με τις απαιτήσεις του μοντέλου, βελτιώνονται οι προβλέψεις.

Πιο συγκεκριμένα, διαβάζει την εικόνα με την χρήση του image_path χρησιμοποιώντας την συνάρτηση cv2.imread. Αλλάζει το μέγεθος της φορτωμένης εικόνας στις διαστάσεις που καθορίσαμε χρησιμοποιώντας τη συνάρτηση cv2.resize του OpenCV. Αυτό διασφαλίζει ότι οι διαστάσεις της εικόνας αντιστοιχούν στο μέγεθος εισόδου που αναμένεται από το μοντέλο βαθιάς μάθησης. Στην συνέχεια κανονικοποιεί τις τιμές των pixels της εικόνας που έχει αλλάξει μέγεθος διαιρώντας κάθε τιμή του pixel με το 255. Αυτό κλιμακώνει τις τιμές των pixels στο εύρος [0, 1], το οποίο χρησιμοποιείται για είσοδο στο νευρωνικό δίκτυο. Τέλος, επιστρέφει την προεπεξεργασμένη εικόνα, η οποία έχει πλέον αλλάξει μέγεθος και έχει κανονικοποιηθεί, με αποτέλεσμα να είναι έτοιμη να χρησιμοποιηθεί για πρόβλεψη από το μοντέλο μηχανικής μάθησης.

```
def preprocess_image(self, image_path):
    img = cv2.imread(image_path)
    img = cv2.resize(img, (Config.IMG_HEIGHT, Config.IMG_WIDTH))
    img = img / 255.0
    return img
```

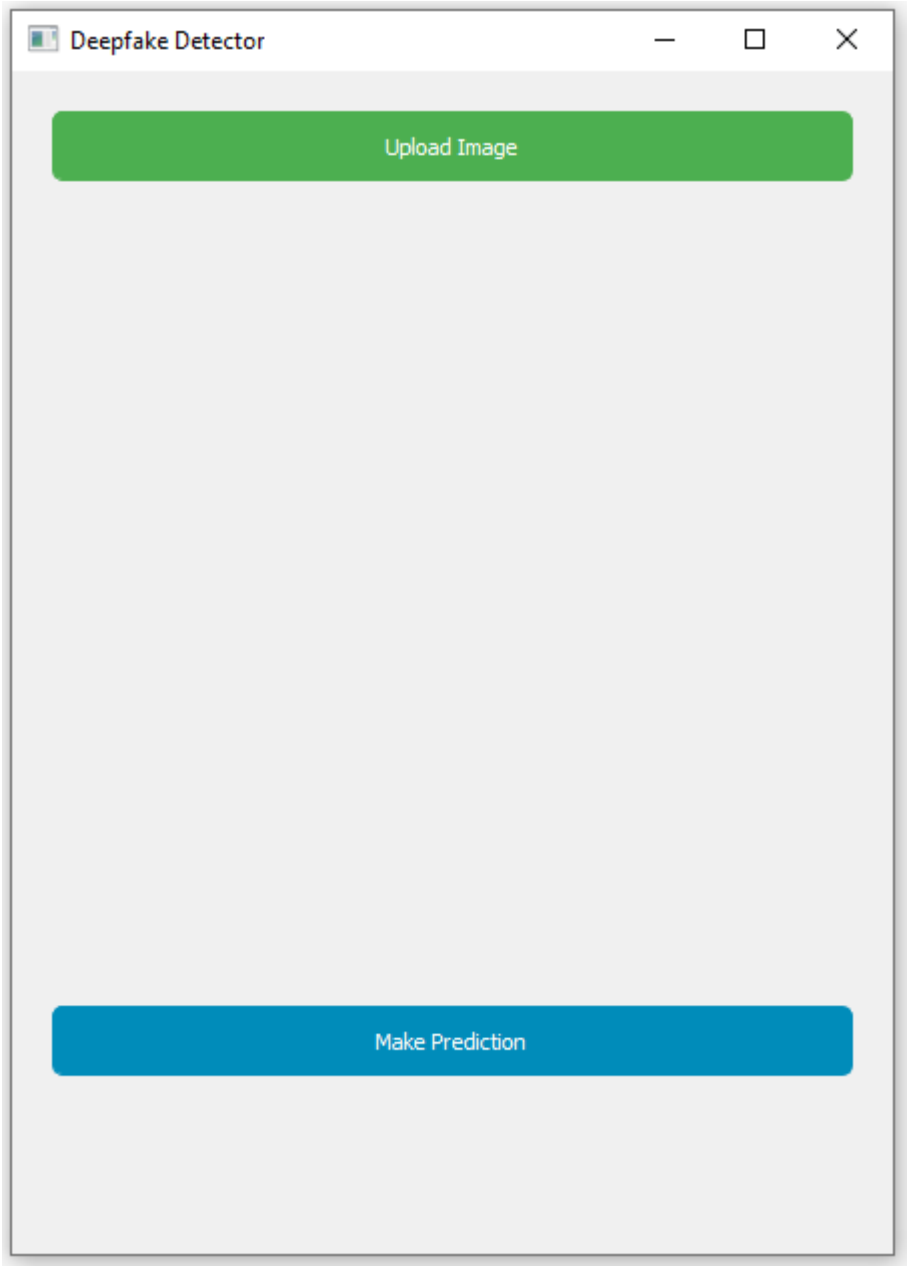
Εικόνα 7.24: Μέθοδος για προετοιμασία εικόνας

7.6 Χρήση της εφαρμογής

Η βασική λειτουργικότητα της εφαρμογής μας περιστρέφεται γύρω από μια φιλική προς τον χρήστη διεπαφή. Ενσωματώνοντας την πολυπλοκότητα της ανίχνευσης των deepfakes σε μια προσιτή διεπαφή, δίνεται η δυνατότητα στους χρήστες να λαμβάνουν τεκμηριωμένες αποφάσεις σχετικά με την ειλικρίνεια του οπτικού περιεχομένου που συναντούν στο διαδίκτυο.

7.6.1 Εκκίνηση της εφαρμογής

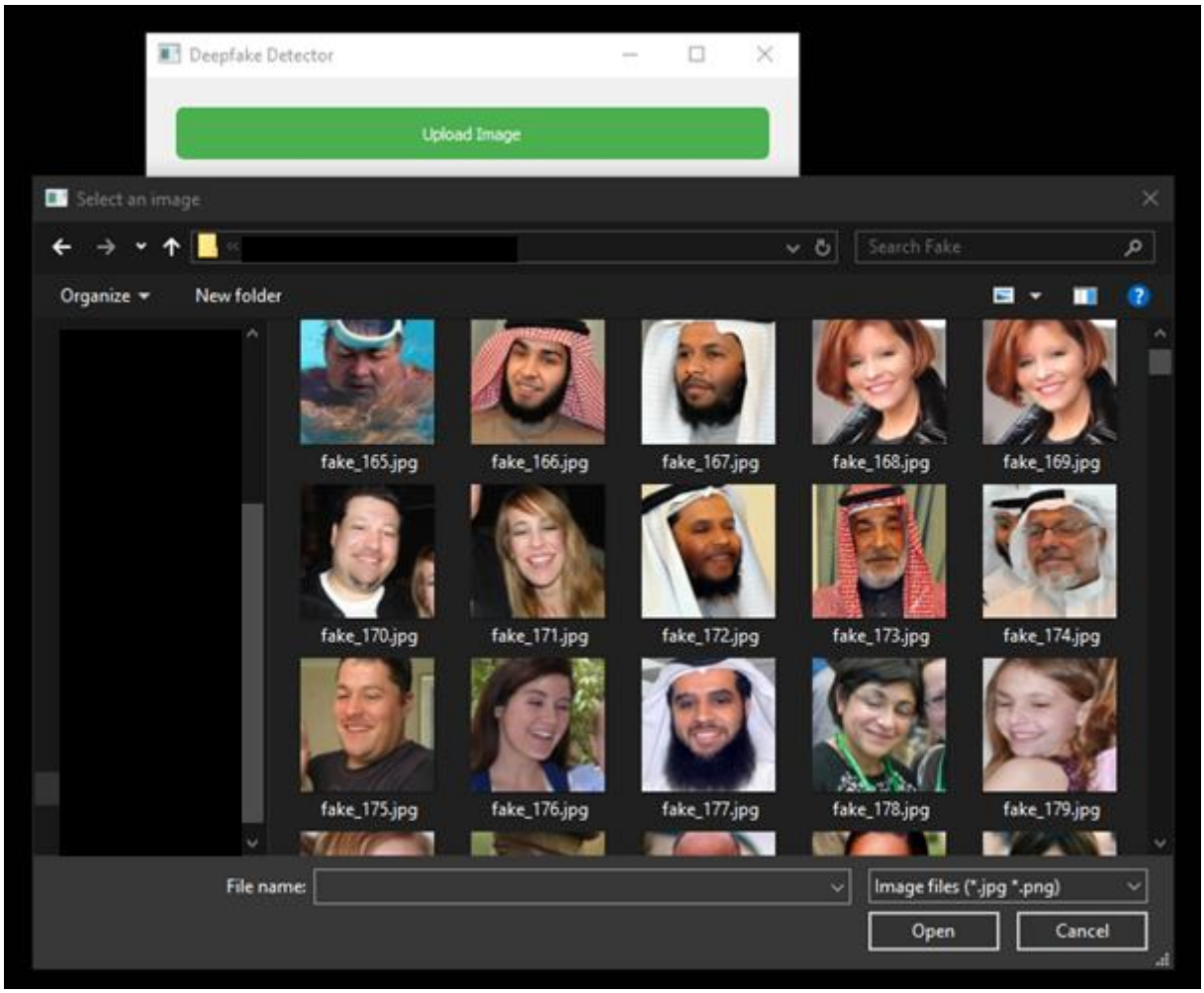
Κατά την εκκίνηση της εφαρμογής, οι χρήστες υποδέχονται μια απλή διάταξη με δυο βασικά κουμπιά, το 'Upload Image ' για το ανέβασμα της εικόνας και το 'Make Prediction' για την δημιουργία της πρόβλεψης.



Εικόνα 7.25: Αρχικό UI εφαρμογής

7.6.2 Επιλογή εικόνας

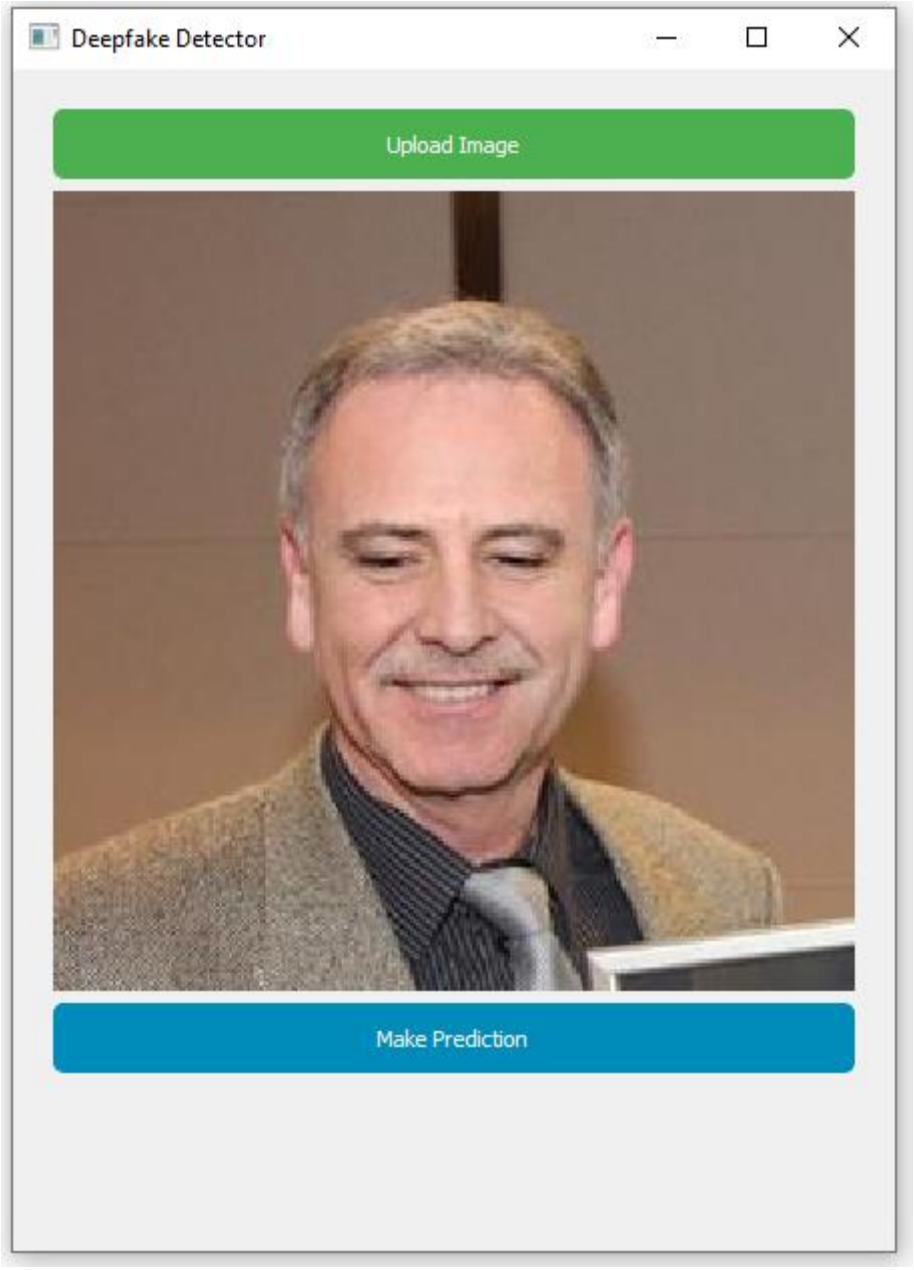
Το κουμπί για το ανέβασμα ενεργοποιεί έναν εξερευνητή αρχείων (file explorer), επιτρέποντας στους χρήστες να επιλέξουν μια εικόνα από την τοπική τους συσκευή.



Εικόνα 7.26: Επιλογή εικόνας

7.6.3 Δημιουργία πρόβλεψης

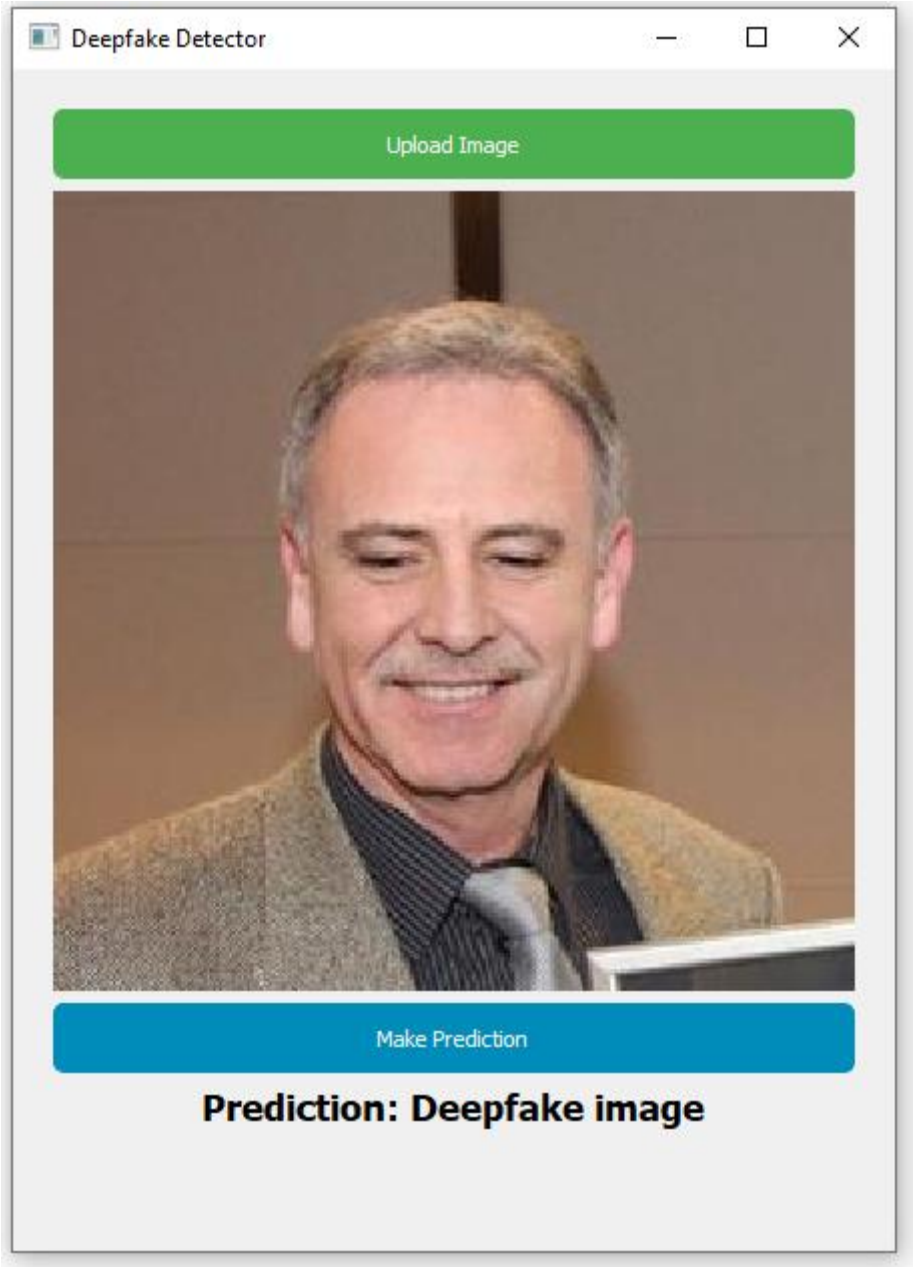
Με την εικόνα να έχει μεταφορτωθεί και να είναι ορατή, οι χρήστες προχωρούν στο επόμενο βήμα πατώντας το κουμπί 'Make Prediction'. Στο παρασκήνιο, η εφαρμογή αξιοποιεί ένα προ-εκπαιδευμένο μοντέλο CNN, το οποίο όπως και είδαμε προηγουμένως έχει εκπαιδευτεί σε ένα ποικίλο σύνολο δεδομένων που περιλαμβάνει τόσο αυθεντικές όσο και deepfake εικόνες.



Εικόνα 7.27: Φόρτωση εικόνας

7.6.4 Αποτέλεσμα πρόβλεψης

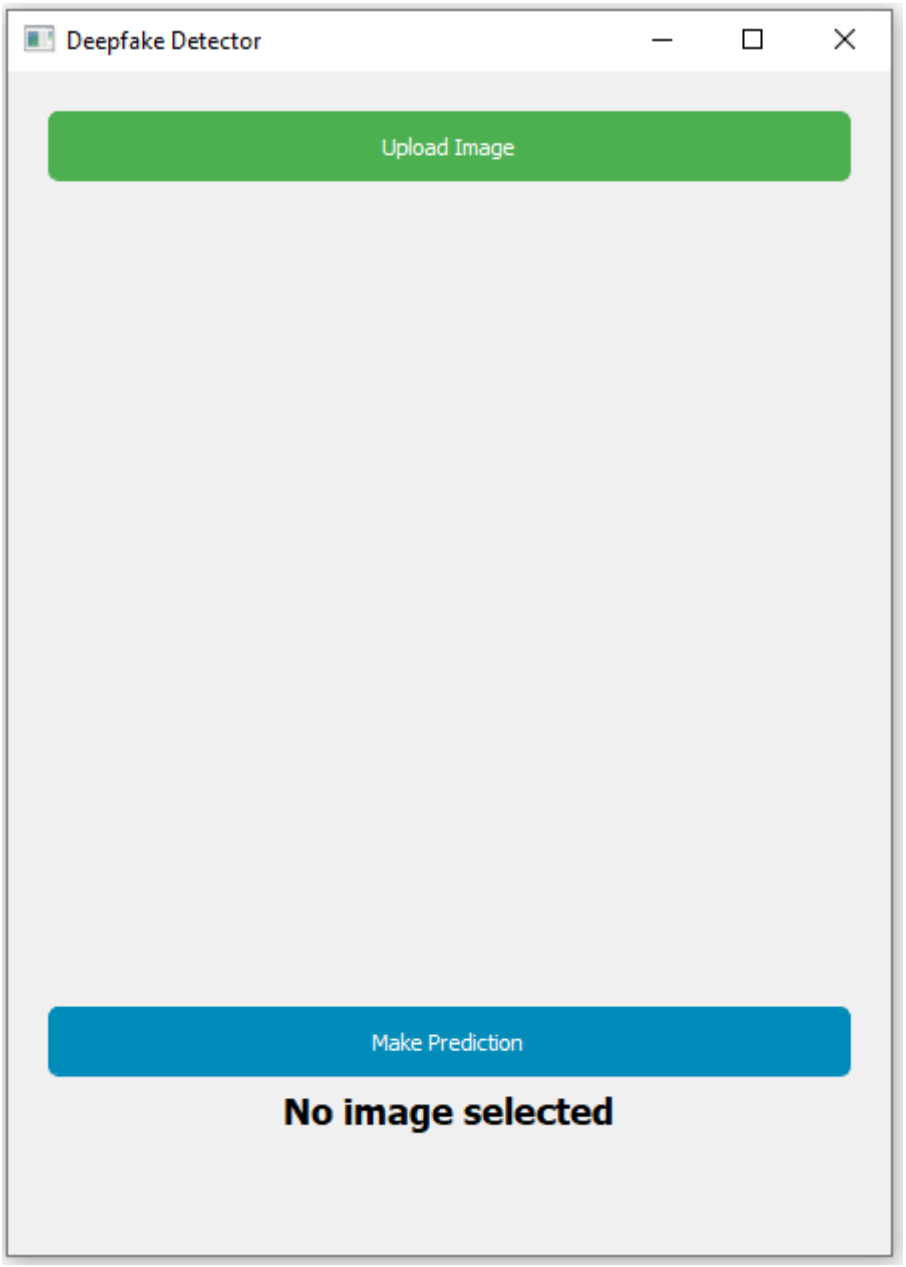
Κατά την κλήση της διαδικασίες πρόβλεψης, οι χρήστες λαμβάνουν αμέσως το αποτέλεσμα απευθείας μέσα από την διεπαφή της εφαρμογής. Εάν η ανεβασμένη εικόνα διαπιστωθεί ότι είναι αυθεντική, η εφαρμογή ενημερώνει τον χρήστη σχετικά, επιβεβαιώνοντας την αξιοπιστία της εικόνας. Αντίθετα, εάν η εικόνα παρουσιάζει χαρακτηριστικά που υποδηλώνουν deepfake, η εφαρμογή ειδοποιεί αμέσως τον χρήστη, προειδοποιώντας τον για πιθανή παραπληροφόρηση.



Εικόνα 7.28: Δημιουργία πρόβλεψης

7.6.5 Μηχανισμός χειρισμού σφαλμάτων

Επιπλέον, η εφαρμογή μας ενσωματώνει μηχανισμούς χειρισμού σφαλμάτων για να εξασφαλίσει μια απρόσκοπτη εμπειρία χρήσης. Σε σενάρια όπου οι χρήστες επιχειρούν να κάνουν μια πρόβλεψη χωρίς να ανεβάσουν μια εικόνα, η εφαρμογή αναχαιτίζει προληπτικά την ενέργεια αυτή, εμφανίζοντας ένα σαφές και συνοπτικό μήνυμα που ενημερώνει τον χρήστη για την παράβλεψη.



Εικόνα 7.29: Χειρισμός σφαλμάτων

7.7 Επίλογος

Συμπερασματικά λοιπόν, σε αυτό το κεφάλαιο παρουσιάστηκε η διαδικασία που ακολουθείται ώστε να αξιοποιηθεί το δημιουργημένο μοντέλο για να πραγματοποιήσουμε μια πρόβλεψη. Αναλύθηκε η διαδικασία της δημιουργίας μια εφαρμογής με γραφικό περιβάλλον, εξηγώντας τόσο το οπτικό όσο και το λειτουργικό κομμάτι. Τέλος, είδαμε πως χρησιμοποιείται η εφαρμογή από τον χρήστη.

Κεφάλαιο 8ο: Το σύνολο δεδομένων

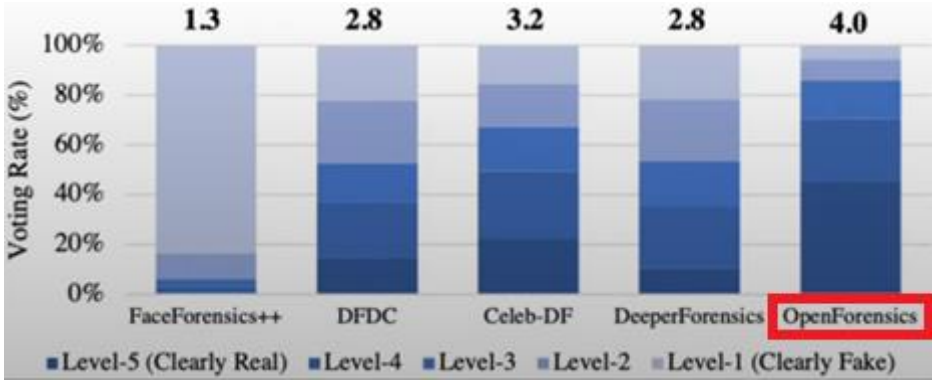
8.1 Εισαγωγή

Παρακάτω θα παρουσιαστεί το σύνολο των δεδομένων (dataset) που χρησιμοποιήθηκε για την εκπαίδευση του μοντέλου καθώς και κάποιες προκλήσεις σχετικά με την χρήση του.

8.2 Το σύνολο των δεδομένων που χρησιμοποιήθηκε

Το dataset που έχει χρησιμοποιηθεί για την πραγματοποίηση της εργασίας είναι το OpenForensics: Large-Scale Challenging Dataset For Multi-Face Forgery Detection And Segmentation In-The-Wild [29]. Αυτό το σύνολο δεδομένων έχει σχεδιαστεί για να δίνει έμφαση στο πρόσωπο, συγκεκριμένα για την ανίχνευση και την κατάτμηση πλαστογραφίας προσώπου. Έτσι, το σύνολο δεδομένων OpenForensics έχει μεγάλες δυνατότητες για την έρευνα τόσο στην αναγνώριση των deepfakes όσο και στην γενική ανίχνευση ανθρώπινων προσώπων.

Η έκδοση 1.0.0 του dataset αποτελείται απο περίπου 115.000 εικόνες οι οποίες περιέχουν περίπου 334.000 ανθρώπινα πρόσωπα. Σε αυτές περιλαμβάνονται οι αυθεντικές εικόνες, αλλά και οι εικόνες που περιέχουν κάποιο deepfake. Το dataset δημιουργήθηκε συλλέγοντας τις αυθεντικές εικόνες από το διαδίκτυο και στην συνέχεια συνθέτοντας τις deepfake εικόνες με την χρήση GAN μοντέλων. Το συνολικό του μέγεθος ήταν 50.8 GB και περιέχει εικόνες με έναν ή περισσότερους ανθρώπους διαφόρων ηλικιών, φύλων, στάσεων, θέσεων, αποκρύψεων προσώπου σε φυσικές σκηνές και με ρεαλιστική εμφάνιση [29]. Το αποτέλεσμα ήταν να δημιουργηθεί ένα ρεαλιστικό και υψηλής ποιότητας dataset.



Εικόνα 8.1: Ρεαλισμός του dataset σύμφωνα με έρευνα [29]

8.3 Προκλήσεις σχετικά με το σύνολο δεδομένων

Υπήρξαν ορισμένες προκλήσεις που αντιμετωπίσαμε με το παραπάνω dataset για τις οποίες κληθήκαμε να βρούμε λύση. Τα προβλήματα είχαν να κάνουν, κυρίως, με τον αριθμό των εικόνων αλλά του συνολικού μεγέθους τους, που έκαναν δύσκολη την διαχείριση του dataset, κυρίως από άποψη

υπολογιστικών πόρων. Επίσης, μας δυσκόλεψε ο τρόπος που διαχωριζόντουσαν οι αυθεντικές με τις deepfake εικόνες, μιας και γινόταν μέσω ενός .json αρχείου και όχι με τον διαχωρισμό τους σε φακέλους. Έτσι, δουλέψαμε μια νέα έκδοση του ίδιου dataset όπου χρησιμοποιήθηκαν περίπου 30.000 από το σύνολο των εικόνων και με τροποποιημένο μέγεθος 256x256. Βοήθησε, επίσης, αρκετά το γεγονός ότι στο τροποποιημένο dataset οι παραποιημένες και οι αυθεντικές φωτογραφίες χωρίζονταν σε διαφορετικούς φακέλους, κάνοντας πιο εύκολο το φόρτωμά τους. Πετύχαμε, έτσι, ένα πιο διαχειρίσιμο σύνολο δεδομένων αξιοποιώντας όμως όση περισσότερη πληροφορία μπορούσαμε να αντλήσουμε το αρχικό.

Κεφάλαιο 9ο: Αξιολόγηση απόδοσης

9.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει μια παρουσίαση των αποτελεσμάτων που προέκυψαν από την εκπαίδευση του CNN για την ταξινόμηση των εικόνων. Καταγράφουμε την δυναμική της εκπαίδευσης σε πολλαπλές εποχές, εξετάζοντας διεξοδικά την εξέλιξη τόσο της ακρίβειας και της απώλειας της εκπαίδευσης όσο και της επικύρωσης. Θα παρουσιαστεί, επίσης, και το πώς επηρεάστηκαν από το learning rate. Προκειμένου να κατανοήσουμε τα αποτελέσματα της εκπαίδευσης, πριν παρουσιαστούν τα γραφήματα, θα εξηγηθούν κάποιες θεωρητικές πληροφορίες σχετικά με τις μετρικές και τις σχέσεις μεταξύ τους. Στόχος είναι να αποκαλύψουμε τους υποκείμενους μηχανισμούς που καθοδηγούν τη διαδικασία μάθησης, διευκρινίζοντας το πώς το μοντέλο αποκτά σταδιακά την ικανότητα να διακρίνει περίπλοκα μοτίβα και χαρακτηριστικά μέσα στα δεδομένα. Μέσω μιας λεπτομερούς ανάλυσης των μετρικών επιδόσεων, για παράδειγμα της ακρίβειας, στοχεύουμε να παρουσιάσουμε μια ολοκληρωμένη αξιολόγηση των δυνατοτήτων του μοντέλου.

9.2 Μετρικές

Οι μετρικές που έχουν καταγραφεί είναι η Απώλεια Εκπαίδευσης (Training Loss), η Ακρίβεια Εκπαίδευσης (Training Accuracy), η Απώλεια Επικύρωσης (Validation Loss) και τέλος η Ακρίβεια Επικύρωσης (Validation Accuracy). Όσον αφορά τις μετρικές που προκύπτουν κατά την διαδικασία της εκπαίδευσης, η Απώλεια Εκπαίδευσης αντιπροσωπεύει το σφάλμα μεταξύ των προβλεπόμενων και των πραγματικών τιμών στα δεδομένα εκπαίδευσης και η Ακρίβεια Εκπαίδευσης μετρά το ποσοστό των σωστά ταξινομημένων περιπτώσεων στα δεδομένα εκπαίδευσης. Σχετικά τις μετρικές που προκύπτουν κατά την διαδικασία της επικύρωσης, η Απώλεια Επικύρωσης αντιπροσωπεύει το σφάλμα του μοντέλου σε ένα ξεχωριστό σύνολο δεδομένων επικύρωσης και η Ακρίβεια Επικύρωσης μετρά το ποσοστό των σωστά ταξινομημένων περιπτώσεων στα δεδομένα επικύρωσης.

Η κατανόηση των σχέσεων μεταξύ των μετρικών μπορεί να βοηθήσει στη διάγνωση προβλημάτων κατά τη διάρκεια της εκπαίδευσης, όπως η υπερπροσαρμογή, η υπεραπλούστευση ή η αστάθεια του μοντέλου. Επιπλέον, μπορεί να καθοδηγήσει τις αποφάσεις ρύθμισης των παραμέτρων για τη βελτιστοποίηση της απόδοσης και της γενίκευσης του μοντέλου.

9.2.1 Απώλεια εκπαίδευσης και απώλεια επικύρωσης

Τυπικά, καθώς μειώνεται η απώλεια εκπαίδευσης, αναμένουμε να μειωθεί και η απώλεια επικύρωσης. Αυτή η σχέση υποδεικνύει ότι το μοντέλο δεν κάνει υπερπροσαρμογή και γενικεύει καλά σε αθέατα δεδομένα. Εάν η απώλεια εκπαίδευσης συνεχίζει να μειώνεται ενώ η απώλεια επικύρωσης αυξάνεται ή παραμένει στάσιμη, αυτό υποδηλώνει υπερπροσαρμογή, όπου το μοντέλο απομνημονεύει τα δεδομένα εκπαίδευσης και αποτυγχάνει να γενικεύσει σε νέα δεδομένα [46].

9.2.2 Ακρίβεια εκπαίδευσης και ακρίβεια επικύρωσης

Παρόμοια με τη σχέση μεταξύ απώλειας εκπαίδευσης και απώλειας επικύρωσης, η αύξηση της ακρίβειας εκπαίδευσης θα πρέπει, ιδανικά, να αντιστοιχεί σε αύξηση της ακρίβειας επικύρωσης. Ένα σημαντικό χάσμα μεταξύ της ακρίβειας εκπαίδευσης και της ακρίβειας επικύρωσης υποδεικνύει υπερπροσαρμογή, όπου το μοντέλο αποδίδει καλά στα δεδομένα εκπαίδευσης αλλά κακώς στα αθέατα δεδομένα [46].

9.2.3 Απώλεια εκπαίδευσης και ακρίβεια εκπαίδευσης

Σε γενικές γραμμές, η μείωση της απώλειας θα πρέπει να συμπίπτει με την αύξηση της ακρίβειας, καθώς οι χαμηλότερες τιμές απώλειας υποδηλώνουν καλύτερη ευθυγράμμιση μεταξύ των προβλεπόμενων και των πραγματικών τιμών. Ωστόσο, είναι δυνατόν η απώλεια να μειώνεται, ενώ η ακρίβεια να βρίσκεται σε οριακό σημείο ή το αντίστροφο, ειδικά σε σενάρια όπου το σύνολο δεδομένων είναι ανισόρροπο ή η συνάρτηση απώλειας δεν ευθυγραμμίζεται απόλυτα με τον στόχο του μοντέλου [46].

9.2.4 Εποχές και σύγκλιση

Η παρατήρηση του τρόπου με τον οποίο οι απώλειες και η ακρίβεια μεταβάλλονται με την πάροδο των εποχών μπορεί να παρέχει πληροφορίες σχετικά με τη συμπεριφορά σύγκλισης του μοντέλου. Μια σταθερή μείωση των απωλειών και αύξηση της ακρίβειας κατά τη διάρκεια των εποχών υποδηλώνει ότι το μοντέλο μαθαίνει σταθερά [46].

9.3 Αξιολόγηση μοντέλου

Παρακάτω θα παρουσιαστούν τα γραφήματα για κάθε μετρική που έχουμε από την διαδικασία της εκπαίδευσης και της επικύρωσης. Θα δούμε έτσι την πρόοδο που είχαμε κατά την εκπαίδευση καθώς και τα τελικά αποτελέσματά της.

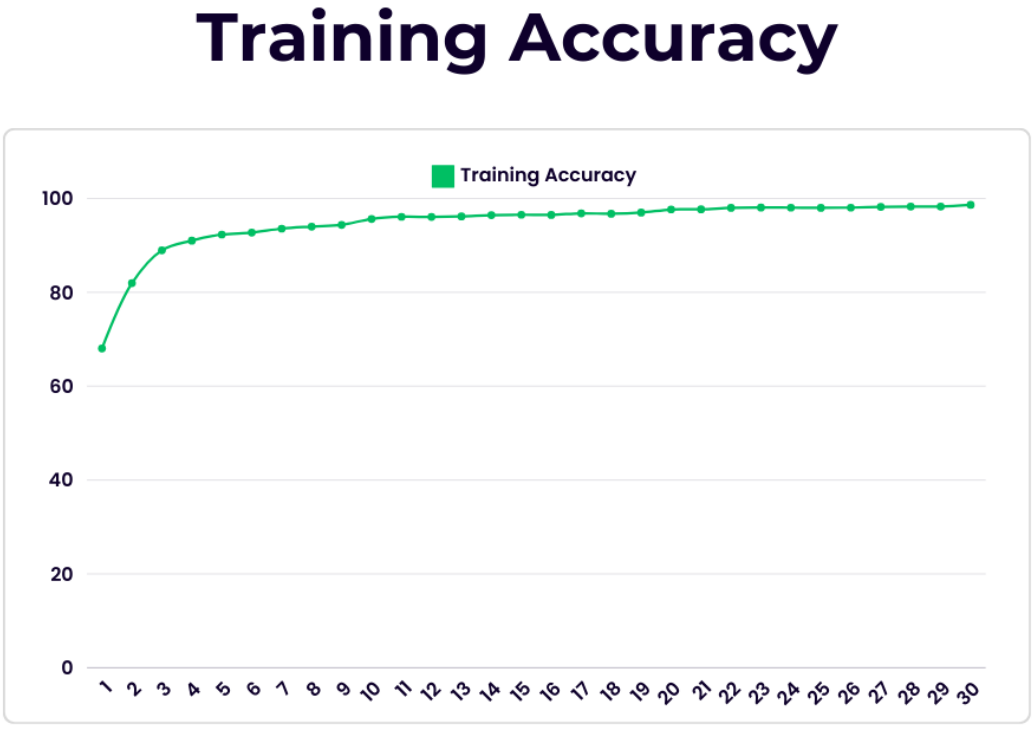
9.3.1 Απώλεια εκπαίδευσης

Το Training Loss μειώνεται σταθερά από 0,6862 σε 0,0378 κατά τη διάρκεια των 30 εποχών. Αυτή η συνεχής μείωση υποδεικνύει ότι το μοντέλο μαθαίνει να ελαχιστοποιεί το σφάλμα του και να βελτιώνει την ακρίβεια πρόβλεψής του. Δεν παρατηρούνται σημαντικές διακυμάνσεις ή ανωμαλίες στην απώλεια εκπαίδευσης, γεγονός που υποδηλώνει μια σταθερή και αποτελεσματική διαδικασία εκπαίδευσης.

Εικόνα 9.1: Απώλεια εκπαίδευσης

9.3.2 Ακρίβεια εκπαίδευσης

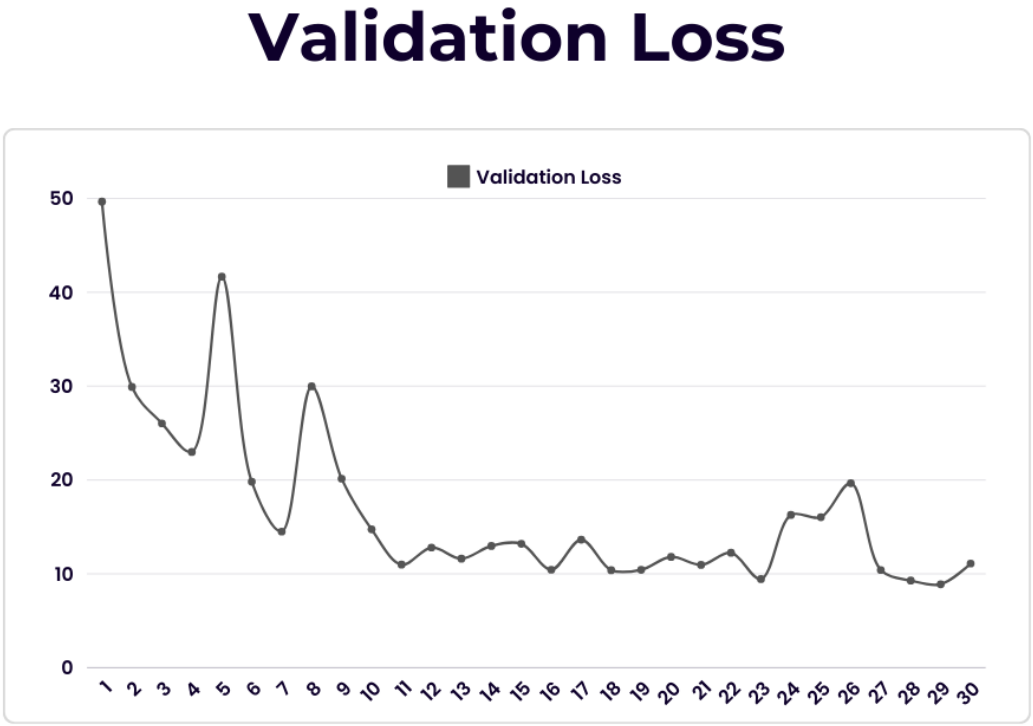
Το Training Accuracy παρουσιάζει σταθερή βελτίωση από 0,6805 σε 0,9862 καθ' όλη τη διάρκεια της εκπαίδευσης. Αυτή η ανοδική τάση υποδηλώνει ότι το μοντέλο μαθαίνει σταδιακά να ταξινομεί με μεγαλύτερη ακρίβεια τα δεδομένα εκπαίδευσης. Όπως και η απώλεια εκπαίδευσης, δεν υπάρχουν αξιοσημείωτες ανωμαλίες ή απότομες αλλαγές στην ακρίβεια εκπαίδευσης, γεγονός που υποδηλώνει μια ομαλή διαδικασία μάθησης.



Εικόνα 9.2: Ακρίβεια εκπαίδευσης

9.3.3 Απώλεια επικύρωσης

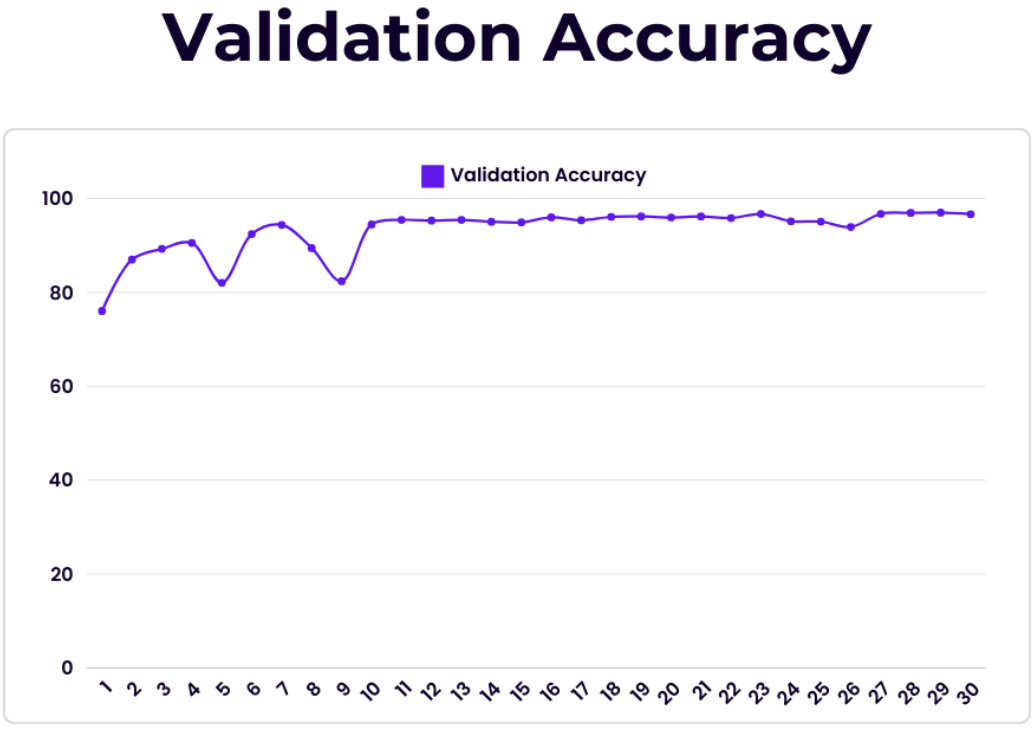
Το Validation Loss αρχικά παρουσιάζει διακυμάνσεις πριν μειωθεί σταδιακά από 0,4966 σε 0,1110 μέχρι το τέλος της εκπαίδευσης. Χάρη στον μηχανισμό model checkpoint callback εμείς αποθηκεύσαμε το καλύτερο μοντέλο που ήταν της εποχής 29 με τιμή 0,0889. Ενώ υπάρχουν διακυμάνσεις, ειδικότερα στην αρχή, που υποδεικνύουν μεταβλητότητα στην απόδοση του μοντέλου σε αθέατα δεδομένα, η συνολική πτωτική τάση από την 10^{η} εποχή και μετά υποδηλώνει ότι το μοντέλο γενικεύει αποτελεσματικά και βελτιώνει την απόδοσή του σε νέα δείγματα.



Εικόνα 9.3: Απώλεια επικύρωσης

9.3.4 Ακρίβεια επικύρωσης

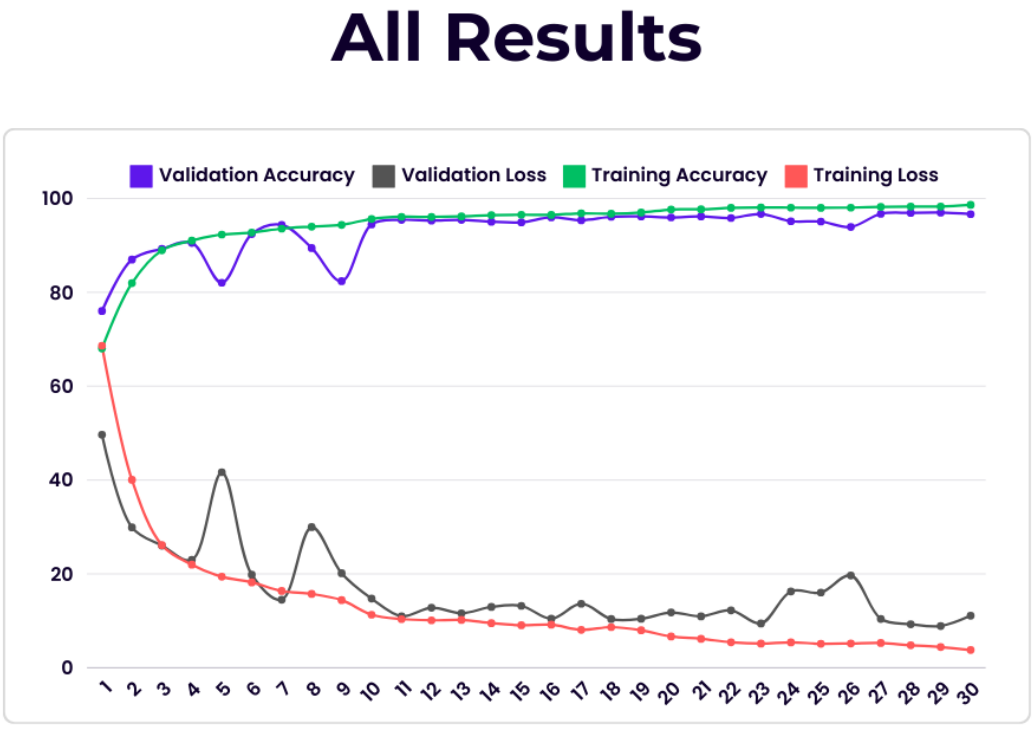
Το Validation Accuracy ξεκινά από 0,7601 και αυξάνεται σημαντικά σε 0,9666 μέχρι την τελευταία εποχή. Χάρη στον μηχανισμό model checkpoint callback εμείς αποθηκεύσαμε το καλύτερο μοντέλο που ήταν της εποχής 29 με τιμή 0,9700. Παρά τις αρχικές διακυμάνσεις, η ακρίβεια επικύρωσης επιδεικνύει συνεχή βελτίωση σε σχέση με τις εποχές εκπαίδευσης, υποδεικνύοντας ότι το μοντέλο μαθαίνει να γενικεύει καλά και να αποδίδει αποτελεσματικά σε αθέατα δεδομένα.



Εικόνα 9.4: Ακρίβεια επικύρωσης

9.3.5 Συνολική απόδοση

Συμπερασματικά λοιπόν, τόσο οι μετρικές εκπαίδευσης όσο και οι μετρικές επικύρωσης παρουσιάζουν θετικές τάσεις, με το μοντέλο να επιδεικνύει σταθερή βελτίωση στις ικανότητες μάθησης και γενίκευσης. Η απουσία σημαντικών ανωμαλιών στη διαδικασία εκπαίδευσης υποδηλώνει ότι το μοντέλο προσαρμόζεται αποτελεσματικά στα δεδομένα και μαθαίνει τα υποκείμενα πρότυπα. Οι αρχικές διακυμάνσεις στις μετρικές επικύρωσης αναδεικνύουν τη μεταβλητότητα της απόδοσης του μοντέλου σε αθέατα δεδομένα κάτι που αργότερα διορθώθηκε με επιτυχία. Συνολικά η υψηλή τελική ακρίβεια επικύρωσης δείχνει ότι το μοντέλο έχει μάθει να γενικεύει αποτελεσματικά και να αποδίδει καλά σε νέα, αθέατα δείγματα, γεγονός που αποτελεί επιθυμητό αποτέλεσμα σε εργασίες μηχανικής μάθησης. Χάρη στον μηχανισμό model checkpoint callback, που όπως είδαμε έχουμε υλοποιήσει, καταφέραμε να αποθηκεύσουμε το καλύτερο μοντέλο που προέκυψε κατά την 29η εποχή, με Validation Loss 0.0889 και Validation Accuracy 0.9700.



Εικόνα 9.5: Συνολική απόδοση

9.4 Ρυθμός μάθησης

Τα αποτελέσματα της εκπαίδευσής σας παρουσιάζουν μια ολοκληρωμένη εικόνα του τρόπου με τον οποίο ο ρυθμός μάθησης επηρεάζει την απόδοση του μοντέλου μας σε 30 εποχές. Το χρονοδιάγραμμα ρυθμού εκμάθησης προσαρμόζεται δυναμικά χρησιμοποιώντας μια κλήση LearningRateScheduler. Ακολουθεί μια ανάλυση του τρόπου με τον οποίο ο ρυθμός μάθησης επηρεάζει τη διαδικασία εκπαίδευσης:

9.4.1 Εποχές 1-9

Στην αρχή της εκπαίδευσης, το μοντέλο ξεκινά με υψηλό ρυθμό μάθησης 0,001. Η ακρίβεια εκπαίδευσης βελτιώνεται σημαντικά από 68,05% στην εποχή 1 σε 94,37% στην εποχή 9. Η ακρίβεια επικύρωσης παρουσιάζει αρχικά σημαντικές βελτιώσεις, φθάνοντας το 94,36% μέχρι την εποχή 7. Ωστόσο, υπάρχουν διακυμάνσεις, με την ακρίβεια επικύρωσης να πέφτει μερικές φορές, για παράδειγμα στο 82,38% στην εποχή 9. Σε αυτήν την φάση, ο υψηλός ρυθμός εκμάθησης επιτρέπει την ταχεία εκμάθηση στα αρχικά στάδια, αλλά προκαλεί και λίγη αστάθεια καθώς το μοντέλο ξεπερνά τις βέλτιστες παραμέτρους, γεγονός που είναι εμφανές από τη διακύμανση της ακρίβειας επικύρωσης.

9.4.2 Εποχές 10-19

Σε αυτό το σημείο ο ρυθμός μάθησης μειώνεται στο μισό σε 0,0005 από την εποχή 10. Η ακρίβεια εκπαίδευσης συνεχίζει να βελτιώνεται, φτάνοντας το 96,17% από την εποχή 13 και σταθεροποιείται γύρω στο 97%. Η ακρίβεια επικύρωσης βελτιώνεται πιο σταθερά και φτάνει στο μέγιστο 96,16% από την εποχή 19. Σε αυτήν την φάση, η μείωση του ρυθμού εκμάθησης βοηθά στη σταθεροποίηση της εκπαίδευσης, επιτρέποντας στο μοντέλο να βελτιώσει τις παραμέτρους του με μεγαλύτερη λεπτομέρεια. Έτσι, αυτή η φάση παρουσιάζει πιο σταθερές βελτιώσεις στην ακρίβεια επικύρωσης σε σύγκριση με την αρχική φάση.

9.4.3 Εποχές 20-29

Σε αυτό το σημείο ο ρυθμός μάθησης μειώνεται περαιτέρω σε 0,00025 από την εποχή 20. Η ακρίβεια εκπαίδευσης φτάνει περίπου στο 98,27% από την εποχή 28, με μικρές διακυμάνσεις. Η ακρίβεια επικύρωσης συνεχίζει να βελτιώνεται, φθάνοντας στο 97,00% από την εποχή 29. Σε αυτήν την φάση, ο χαμηλότερος ρυθμός μάθησης επιτρέπει ακόμη πιο λεπτομερείς προσαρμογές στις παραμέτρους του μοντέλου, γεγονός που βελτιώνει τόσο την ακρίβεια εκπαίδευσης όσο και την ακρίβεια επικύρωσης. Ωστόσο, τα κέρδη γίνονται οριακά, γεγονός που υποδηλώνει ότι το μοντέλο πλησιάζει τη βέλτιστη απόδοσή του.

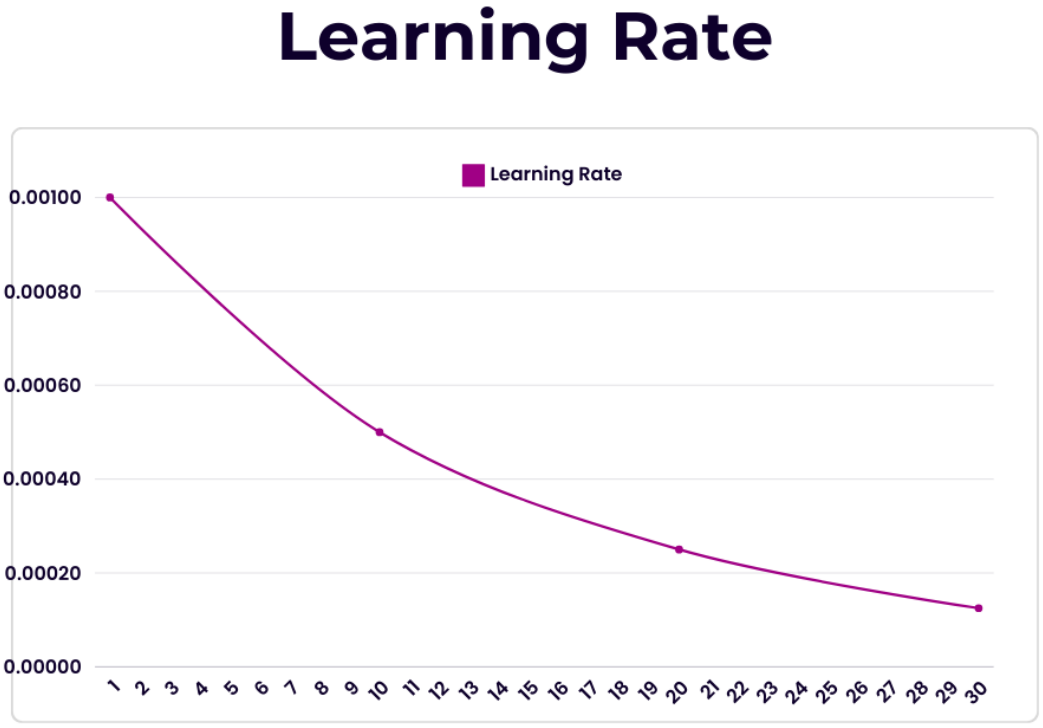
9.4.4 Εποχή 30

Σε αυτό το σημείο ο ρυθμός μάθησης μειώνεται και πάλι στο μισό σε 0,000125 για την τελευταία εποχή. Η ακρίβεια της εκπαίδευσης είναι 98,62%, αλλά η ακρίβεια επικύρωσης είναι ελαφρώς χαμηλότερη στο 96,66%. Ο πολύ χαμηλός ρυθμός μάθησης σε αυτό το στάδιο βοηθά στην πραγματοποίηση πολύ λεπτών προσαρμογών. Ωστόσο, όπως παρατηρήθηκε, η ακρίβεια επικύρωσης δεν βελτιώνεται, γεγονός που υποδηλώνει φθίνουσα απόδοση από την περαιτέρω μείωση του ρυθμού μάθησης.

9.4.5 Συμπεράσματα για ρυθμό μάθησης

Η αρχική φάση (υψηλός ρυθμός μάθησης), επιτρέπει τη γρήγορη σύγκλιση, αλλά μπορεί να προκαλέσει αστάθεια και υπέρβαση, ιδιαίτερα ορατή στις πρώτες εποχές, όπου η ακρίβεια επικύρωσης παρουσιάζει διακυμάνσεις. Η μέση φάση (μέτριος ρυθμός μάθησης) παρέχει ισορροπία, οδηγώντας σε σταθερή και συνεπή βελτίωση τόσο της ακρίβειας εκπαίδευσης, όσο και της επικύρωσης. Η αργότερη φάση (χαμηλός ρυθμός μάθησης) διευκολύνει τη λεπτομερή ρύθμιση, με αποτέλεσμα οριακές βελτιώσεις, υποδεικνύοντας ότι το μοντέλο φτάνει στη βέλτιστη απόδοση. Η τελική φάση πραγματοποιεί τις τελικές προσαρμογές που είναι χρήσιμες για τη λεπτομερή ρύθμιση, αλλά με περιορισμένο αντίκτυπο στην ακρίβεια επικύρωσης, υποδηλώνοντας ότι πλησιάζει τις βέλτιστες παραμέτρους του μοντέλου.

Συνολικά, το χρονοδιάγραμμα ρυθμού μάθησης φαίνεται να είναι καλά βαθμονομημένο, ξεκινώντας με υψηλό ρυθμό μάθησης για τη γρήγορη εκμάθηση των γενικών μοτίβων και μειώνοντάς τον προοδευτικά για τη λεπτομερή ρύθμιση των παραμέτρων του μοντέλου, εξασφαλίζοντας τόσο σταθερότητα όσο και υψηλή απόδοση σε μεταγενέστερες εποχές. Αυτή η προσέγγιση βελτίωσε αποτελεσματικά την ακρίβεια του μοντέλου μας σε ένα αξιέπαινο 97,00% στο σύνολο επικύρωσης.



Εικόνα 9.6: Εξέλιξη του learning rate

9.5 Επίλογος

Σε αυτό το κεφάλαιο, αφού κατανοήθηκαν οι μετρικές που αξιολογούν το μοντέλο μας καθώς και οι σχέσεις μεταξύ τους, παρουσιάσαμε τα αποτελέσματα τόσο της διαδικασίας της εκπαίδευσης όσο και την διαδικασία της επικύρωσης καθώς και το πώς επηρεάστηκαν από το learning rate. Ως εκ τούτου, πραγματοποιήσαμε μια ολοκληρωμένη αξιολόγηση των δυνατοτήτων του μοντέλου.

Κεφάλαιο 10ο: Συμπεράσματα, μελλοντικές προτάσεις και προκλήσεις

10.1 Εισαγωγή

Στο παρόν κεφάλαιο θα αναφέρουμε μερικά συμπεράσματα που αντλήσαμε κατά την δημιουργία της παρούσας διπλωματικής καθώς και μερικές προτάσεις για μελλοντικές βελτιώσεις που μπορούν να πραγματοποιηθούν αλλά και προκλήσεις που καλούμαστε να αντιμετωπίσουμε.

10.2 Συμπεράσματα

Η αναγνώριση των deepfakes είναι μια σημαντική πρόκληση που καλούμαστε να αντιμετωπίσουμε στην συνεχόμενη μάχη που δίνουμε για την ακεραιότητα των ψηφιακών πληροφοριών. Η ταχεία εξέλιξη της μηχανικής μάθησης βοηθάει στην υλοποίηση εργαλείων ώστε να πετύχουμε τον στόχο μας.

Στην παρούσα εργασία αναπτύξαμε ένα μοντέλο μηχανικής μάθησης, όπου μετά από αρκετές προσαρμογές και δοκιμές, έφτασε σε ένα σημείο που το καθιστά χρήσιμο εργαλείο για να πετύχουμε τον σκοπό μας. Μέσω πειραματισμού και ανάλυσης, αποδείξαμε την αποτελεσματικότητα του μοντέλου μας στον ακριβή εντοπισμό βίντεο deepfake, συμβάλλοντας έτσι στις ευρύτερες προσπάθειες για την καταπολέμηση της παραπληροφόρησης και τη διατήρηση της ακεραιότητας του ψηφιακού περιεχομένου. Τα ευρήματά μας υπογραμμίζουν τη σημασία της αξιοποίησης προηγμένων τεχνικών μηχανικής μάθησης, όπως τα CNN, στη μάχη κατά των εξελισσόμενων απειλών που θέτουν τα συνθετικά μέσα ενημέρωσης.

10.3 Μελλοντικές βελτιώσεις

Καθώς ολοκληρώνουμε αυτή την εργασία, πρέπει να αναγνωρίσουμε τις πιθανές οδούς για μελλοντικές βελτιώσεις στο μοντέλο ανίχνευσης deepfakes. Ενώ το μοντέλο μας επιδεικνύει ελπιδοφόρα αποτελέσματα, πάντα υπάρχουν περιθώρια βελτίωσης. Πρώτον, σε προσπάθειες όπου υπάρχει διαθέσιμη μεγαλύτερη υπολογιστική ισχύ θα μπορούσε να διαφοροποιηθεί και να αυξηθεί ακόμα παραπάνω το σύνολο δεδομένων κάτι που θα βοηθούσε στην ενίσχυση της ικανότητας του μοντέλου να γενικεύει σε διάφορα σενάρια του πραγματικού κόσμου, που περιλαμβάνουν ένα ευρύτερο φάσμα εκφράσεων προσώπου, συνθηκών φωτισμού και φόντου. Δεύτερον, η διερεύνηση εναλλακτικών αρχιτεκτονικών ή η ενσωμάτωση πρόσθετων επιπέδων, θα μπορούσε ενδεχομένως να ενισχύσει την ευαισθησία του μοντέλου σε λεπτές ενδείξεις που υποδηλώνουν deepfake. Επιπλέον, θα μπορούσε να γίνει ενσωμάτωση τεχνικών αντιπαραθετικής εκπαίδευσης (Adversarial training) για την αντιμετώπιση των στρατηγικών που χρησιμοποιούνται συνήθως από κακόβουλους φορείς για να αποφύγουν την ανίχνευση. Η βελτιστοποίηση σε πραγματικό χρόνο για συσκευές και πλατφόρμες με περιορισμένους πόρους, σε συνδυασμό με ακόμα πιο φιλικές προς τον χρήστη διεπαφές για ευρύτερη προσβασιμότητα, θα μπορούσε να ενισχύσει την πρακτική χρησιμότητα του μοντέλου μας. Τέλος, η συνεχής αξιολόγηση και συγκριτική αξιολόγηση σε σχέση με εξελισσόμενα σύνολα δεδομένων και σύγχρονες προσεγγίσεις είναι υψίστης σημασίας για να διασφαλιστεί ότι το μοντέλο μας παραμένει ανταγωνιστικό και

αποτελεσματικό στο δυναμικό τοπίο των deepfakes. Παρόλο που το μοντέλο μας αποτελεί ένα βήμα προς τα εμπρός, η υιοθέτηση αυτών των οδών βελτίωσης θα είναι απαραίτητη στη συνεχή επιδίωξή μας να καταπολεμήσουμε την παραπληροφόρηση και να διατηρήσουμε την ακεραιότητα του ψηφιακού περιεχομένου.

10.4 Προκλήσεις

Καθώς προχωράμε μπροστά, η αντιμετώπιση της όλο και εξελισσόμενης τεχνολογίας των deepfakes παρουσιάζει μια πληθώρα προκλήσεων που απαιτούν συνεχή έρευνα και καινοτομία. Οι προκλήσεις αυτές αφορούν τόσο την τεχνολογία όσο και την ηθική.

10.1 Τεχνολογικές προκλήσεις

Η πρόοδος των τεχνολογιών ανίχνευσης deepfakes αντιμετωπίζει πλήθος τεχνικών προκλήσεων. Μια από τις σημαντικότερες προκλήσεις έγκειται στη δυναμική φύση των τεχνικών για την δημιουργία deepfakes, οι οποίες εξελίσσονται συνεχώς σε επίπεδο πολυπλοκότητας και προσαρμοστικότητας. Όσο τελειοποιούνται οι μέθοδοι για να δημιουργούνται όλο και πιο ρεαλιστικές παραποιήσεις, η δημιουργία όλο και πιο ισχυρών αλγορίθμων ανίχνευσης γίνεται πιο δύσκολο. Επιπλέον, ο τεράστιος όγκος και η ποικιλομορφία του ψηφιακού περιεχομένου που κυκλοφορεί στο διαδίκτυο θέτουν σημαντικά εμπόδια για τα συστήματα ανίχνευσης απομίμησης, καθιστώντας αναγκαία την ύπαρξη λύσεων ικανών να επεξεργάζονται τεράστιες ποσότητες δεδομένων σε πραγματικό χρόνο.

Επιπλέον, το "κυνηγητό" μεταξύ των δημιουργών deepfakes και των δημιουργών αλγορίθμων ανίχνευσης των deepfakes δημιουργεί μια διαρκή κούρσα, όπου κάθε πρόοδος στις τεχνικές ανίχνευσης μπορεί να ωθήσει τους αντιπάλους να επινοήσουν νέες στρατηγικές αποφυγής και το αντίστροφο. Αυτό καθιστά αναγκαίο να υπάρχει ένας συνεχής κύκλος έρευνας και καινοτομίας για την παρακολούθηση των αναδυόμενων απειλών και τη διατήρηση της αποτελεσματικότητας των συστημάτων ανίχνευσης.

Τέλος, ο απαιτητικός σε πόρους χαρακτήρας των μοντέλων βαθιάς μάθησης, σε συνδυασμό με την ανάγκη για μεγάλης κλίμακας σύνολα δεδομένων για εκπαίδευση, παρουσιάζει πρακτικούς περιορισμούς που μπορεί να περιορίσουν την ανάπτυξη συστημάτων ανίχνευσης deepfakes, ιδίως σε περιβάλλοντα με περιορισμένους πόρους.

10.2 Ηθικές προκλήσεις

Η ανίχνευση των deepfakes μπορεί να αντιμετωπίσει και μια πληθώρα από προκλήσεις, οι οποίες υπερβαίνουν την τεχνική πολυπλοκότητα. Οι προκλήσεις αυτές αφορούν πρωτίστως τις ηθικές ανησυχίες που απορρέουν από τη διάδοση και την κακή χρήση των τεχνολογιών deepfake. Καθώς οι δυνατότητες των αλγορίθμων deepfake συνεχίζουν να εξελίσσονται, η δυνατότητα εκμετάλλευσής τους σε διάφορα κακόβουλα πλαίσια, συμπεριλαμβανομένης της παραπληροφόρησης, της προπαγάνδας και της πλαστοπροσωπίας, εγείρει σημαντικές ανησυχίες. Οι ηθικές επιπτώσεις που αφορούν την προστασία της ιδιωτικής ζωής και της συναίνεσης είναι ιδιαίτερα σημαντικές, καθώς η τεχνολογία deepfake μπορεί να χρησιμοποιηθεί για τη χειραγώγηση και την κατασκευή της εικόνας ατόμων χωρίς τη γνώση ή τη συναίνεσή τους, οδηγώντας σε βαθιές επιπτώσεις στην προσωπική και κοινωνική εμπιστοσύνη.

Επιπλέον, ο εκδημοκρατισμός των εργαλείων deepfake και η προσβασιμότητά τους σε άτομα με κακόβουλες προθέσεις κάνει ακόμα πιο αναγκαία την αντιμετώπιση των ηθικών προβληματισμών και της εφαρμογής δικλείδων ασφαλείας για τον μετριασμό των πιθανών βλαβών. Η δημιουργία ισορροπίας μεταξύ της προώθησης της καινοτομίας καθώς και η τήρηση των αρχών της ιδιωτικής ζωής, της συναίνεσης και της ψηφιακής ακεραιότητας θα απαιτήσει διεπιστημονική συνεργασία, ρυθμιστικά πλαίσια και συνεχή διάλογο εντός της ερευνητικής κοινότητας και γενικότερα της κοινωνίας.

Τέλος, το νομικό τοπίο που περιβάλλει την τεχνολογία deepfake, συμπεριλαμβανομένων των ζητημάτων ευθύνης, απόδοσης και δικαιοδοσίας, παρουσιάζει τεράστιες προκλήσεις που χρήζουν προσεκτικής εξέτασης.

10.3 Επίλογος

Στο κεφάλαιο αυτό μιλήσαμε για τα συμπεράσματα που αντλήσαμε κατά την διεκπεραίωση αυτής της διπλωματικής εργασίας καθώς και για κάποιες μελλοντικές βελτιώσεις που θα μπορούσαν να γίνουν πάνω σε αυτήν. Τέλος, τονίσαμε την ανάγκη για την επίλυση τόσο κάποιων τεχνολογικών όσο και ηθικών προκλήσεων που θα κληθούμε να αντιμετωπίσουμε στο μέλλον.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] R. C. Gonzalez, R. E. Woods, and B. R. Masters, "Digital Image Processing, Third Edition," *Journal of Biomedical Optics*, vol. 14, no. 2, p. 029901, 2009, doi: https://doi.org/10.1117/1.3115362.

[2] Mr. A. K, Miss. A. K, Mr. D. N, and Miss. D. S. M K, "Detecting Fake Images Using Machine Learning," International Journal of Research Publication and Reviews, vol. 4, no. 4, pp. 2063–2069, Apr. 2023, doi: https://doi.org/10.55248/gengpi.2023.4.4.35702.

[3] N. Kumar and Toshanlal Meenpal, "Semantic Segmentation-Based Image Inpainting Detection," *Lecture notes in electrical engineering*, pp. 665–677, Jul. 2020, doi: https://doi.org/10.1007/978-981-15-4692-1_51.

[4] Jeff Da, Maxwell Forbes, Rowan Zellers, Anthony Zheng, Jena D. Hwang, Antoine Bosselut, and Yejin Choi. 2021. Edited Media Understanding Frames: Reasoning About the Intent and Implications of Visual Misinformation.

[5] P. L. Stanchev, D. Green Jr. and B. Dimitrov. "High level colour similarity retrieval", International Journal of Information Theories and Applications, vol. 10, no. 3, (2003), pp. 363-369.

[6] J. L. Spudich and B. H. Satir, Sensory Receptors and Signal Transduction. New York: Wiley-Liss, 2001.

[7] A. K. Jain and A. Vailaya, "Image retrieval using colour and shape", Pattern Recognition, vol. 29, no. 8, (1996), pp. 1233-1244.

[8] S. Pashine, S. Mandiya, P. Gupta, and R. Sheikh, "Deep Fake Detection: Survey of Facial Manipulation Detection Solutions," Jun. 2021.

[9] D. Tian, "A Review on Image Feature Extraction and Representation Techniques," Multimedia and Ubiquitous Engineering, vol. 8, no. 4, pp. 385–396, Jul. 2013.

[10] G. Pass and R. Zabith, "Histogram refinement for content-based image retrieval", In Proc. Workshop on Applications of Computer Vision, (1996), pp. 96-102.

[11] M. Flickner, H. Sawhney, W. Niblack, et al., "Query by image and video content: the QBIC system", IEEE Computer, vol. 28, no. 9, (1995), pp. 23-32.

[12] J. Huang, S. Kuamr, M. Mitra, et al., "Image indexing using colour correlogram", In Proc. CVPR, (1997), pp. 762-765.

[13] Ray, "Color, Shape and Texture: Feature Extraction using OpenCV," *Medium*, Jan. 26, 2023. https://raychunyin00.medium.com/color-shape-and-texture-feature-extraction-using-opencv-cb1feb2dbd73 (accessed May 19, 2024).

[14] C. C. Hung, E. Song, Y. Lan, and Springerlink (Online Service, *Image Texture Analysis : Foundations, Models and Algorithms*. Cham: Springer International Publishing, 2019.

[15] "Texture Classification," *apmonitor.com*. https://apmonitor.com/pds/index.php/Main/TextureClassification (accessed May 19, 2024).

[16] D. Zhang and G. Lu, "Review of shape representation and description techniques", Pattern Recognition, vol. 37, no. 1, (2004), pp. 1-19.

[17] C. Yang, M. Dong and F. Fotouhi, “Image content annotation using Bayesian framework and complement components analysis”, In Proc. ICIP, (2005). International Journal of Multimedia and Ubiquitous Engineering Vol. 8, No. 4, July, 2013 395.

[18] V. Mezaris, I. Kompatsiaris and M. G. Strintzis, “An ontology approach to object-based image retrieval”, In Proc. ICIP, (2003), pp. 511-514.

[19] D. Zhang, M. M. Islam, G. Lu, et al., “Semantic image retrieval using region based inverted file”, In Proc. DICTA, (2009), pp. 242-249.

[20] D. C. MAVUZER, “Canny Edge Detection Algorithm with Python,” *Medium*, Apr. 22, 2022. https://medium.com/@ceng.mavuzer/canny-edge-detection-algorithm-with-python-17ac62c61d2e.

[21] R. Chesney and D. Citron, "Deepfakes and the new disinformation war: The coming age of post-truth geopolitics," Foreign Affairs, vol. 98, no. 1, Jan./Feb. 2019.

[22] Temir, Erkam. (2020). Deepfake: New Era in The Age of Disinformation & End of Reliable Journalism. 10.18094/JOSC.685338.

[23] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, “MesoNet: a Compact Facial Video Forgery Detection Network,” *arXiv:1809.00888 [cs, eess]*, Sep. 2018, Available: https://arxiv.org/abs/1809.00888.

[24] J. Negoita, “Deepfake AI Face Swap,” *Medium*, Jan. 04, 2024. https://medium.com/@codingdudecom/deepfake-ai-face-swap-e11edc7d67e1 (accessed May 19, 2024).

[25] W. Zhao and Rama Chellappa, Face Processing: Advanced Modeling and Methods. Elsevier, 2011.

[26] I. Kliashchou, “Face Morphing: Learn How to Protect Against Identity Fraud,” *Regula*, Mar. 29, 2023. https://regulaforensics.com/blog/facial-morphing/ (accessed May 19, 2024).

[27] B. Ferwerda, M. Schedl, and M. Tkalcic, “Predicting Personality Traits with Instagram Pictures,” Proceedings of the 3rd Workshop on Emotions and Personality in Personalized Systems 2015 - EMPIRE ’15, 2015, doi: https://doi.org/10.1145/2809643.2809644.

[28] K. P. Murphy, *Machine Learning : a Probabilistic Perspective*. Cambridge (Ma): Mit Press, 2012.

[29] T.-N. Le, H. H. Nguyen, and J. Yamagishi, “OpenForensics: Multi-Face Forgery Detection And Segmentation In-The-Wild Dataset [V.1.0.0],” zenodo.org, Oct. 2021, doi: https://doi.org/10.5281/zenodo.5528418.

[30] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-scale Image Recognition,” 2015.

[31] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” Nature, vol. 521, no. 7553, pp. 436–444, May 2015, doi: https://doi.org/10.1038/nature14539.

[32] C. Camacho, “Convolutional Neural Networks,” *cezannec.github.io*, Jun. 03, 2013. https://cezannec.github.io/Convolutional_Neural_Networks/.

[33] M. Abadi et al., “TensorFlow: A System for Large-Scale Machine Learning TensorFlow: A system for large-scale machine learning,” pp. 265–283, Nov. 2016.

[34] Keras, “Keras documentation: About Keras,” keras.io. https://keras.io/about/.

[35] M. Sharma, “Generalization in Machine Learning for better performance.,” Medium, Apr. 29, 2019. https://mathanrajsharma.medium.com/generalization-in-machine-learning-for-better-performance-51bed74a3820.

[36] S. Saxena, “Overfitting And Underfitting in Machine Learning,” Analytics Vidhya, Feb. 07, 2020. https://www.analyticsvidhya.com/blog/2020/02/underfitting-overfitting-best-fitting-machine-learning/.

[37] Z. Keita, “Classification in Machine Learning: A Guide for Beginners,” Datacamp, Sep. 2022. https://www.datacamp.com/blog/classification-machine-learning.

[38] “Why Should the Data Be Shuffled for Machine Learning Tasks?,” *GeeksforGeeks*, Feb. 14, 2024. https://www.geeksforgeeks.org/why-should-the-data-be-shuffled-for-machine-learning-tasks/.

[39] A. L. Duca, “Understanding the Difference between Training, Test, and Validation Sets in Machine Learning,” syntax-error, Jun. 16, 2023. https://medium.com/syntaxerrorpub/understanding-the-difference-between-training-test-and-validation-sets-in-machine-learning-c59feec6483b.

[40] A. A. Awan, “A Complete Guide to Data Augmentation,” *www.datacamp.com*, Nov. 2022. https://www.datacamp.com/tutorial/complete-guide-data-augmentation.

[41] D.-M. L. M. Simple, “How does Batch Size impact your model learning,” Geek Culture, Feb. 05, 2022. https://medium.com/geekculture/how-does-batch-size-impact-your-model-learning-2dd34d9fb1fa.

[42] Simplilearn, “What is Epoch in Machine Learning? | Simplilearn,” *Simplilearn.com*, Aug. 30, 2022. https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-epoch-in-machine-learning.

[43] T. Martin, “A (Very Short) Visual Introduction to Learning Rate Schedulers (With Code),” *Medium*, Jul. 09, 2023. https://medium.com/@theom/a-very-short-visual-introduction-to-learning-rate-schedulers-with-code-189eddffdb00 (accessed May 15, 2024).

[44] R. C. Limited, “PyQt5: Python bindings for the Qt cross platform application toolkit,” PyPI. https://pypi.org/project/PyQt5/.

[45] “Qt Style Sheets | Qt Widgets 6.7.1,” *doc.qt.io*. https://doc.qt.io/qt-6/stylesheet.html (accessed May 01, 2024).

[46] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, Massachusetts: The Mit Press, 2016. Available: http://imlab.postech.ac.kr/dkim/class/csed514_2019s/DeepLearningBook.pdf.