



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΥΠΟΛΟΓΙΣΤΩΝ

ΡΟΗ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Classification and style transfer of animation using deep neural networks

**Ταξινόμηση και μεταφορά τεχνοτροπίας (στυλ) κινουμένων σχεδίων με
χρήση βαθέων νευρωνικών δικτύων**

Διπλωματική Εργασία

Χατζηλίας Κωνσταντίνος

Επιβλέπων : Αθανάσιος Βουλόδημος,

Επίκουρος Καθηγητής ΠΑ.Δ.Α

Αθήνα, Ιούλιος 2021



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΥΠΟΛΟΓΙΣΤΩΝ

ΡΟΗ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Classification and style transfer of animation using deep neural networks

Ταξινόμηση και μεταφορά τεχνοτροπίας (στυλ) κινουμένων σχεδίων με
χρήση βαθέων νευρωνικών δικτύων

Διπλωματική Εργασία

Χατζηλίας Κωνσταντίνος

Επιβλέπων : Αθανάσιος Βουλόδημος,

Επίκουρος Καθηγητής ΠΑ.Δ.Α

.....
Αθανάσιος Βουλόδημος,
Επ. Καθηγητής

.....
Γεώργιος Μπαρδής,
Επ. Καθηγητής

.....
Παναγιώτα Τσελέντη,
ΕΔΙΠ

Αθήνα, Ιούλιος 2021

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Χατζηγίας Κωνσταντίνος του Θεοδοσίου, με αριθμό μητρώου 151091, φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της πτυχιακής/διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών



Περίληψη

Η αναγνώριση και η ταξινόμηση εικόνας είναι οι δύο πιο συχνές περιπτώσεις χρήσης των CNN στον τομέα της Μηχανικής Μάθησης. Πηγαίνοντας ένα βήμα παραπέρα, η ανίχνευση και ταξινόμηση διαφορετικών στυλ, καθώς και η μεταφορά ενός στυλ, από μια εικόνα σε μια άλλη είναι μια ακόμα ενδιαφέρουσα εφαρμογή των CNN.

Ο στόχος είναι η δημιουργία ενός μοντέλου το οποίο, λαμβάνοντας υπόψη αρκετές εικόνες διαφορετικών στυλ (είτε πρόκειται για διαφορετικά στυλ τέχνης, στυλ κινούμενων σχεδίων κλπ), αναγνωρίζει, διαφοροποιεί και προβλέπει με ακρίβεια το στυλ μιας εικόνας. Για το δεύτερο μέρος, χρησιμοποιώντας τμήματα ενός μοντέλου, μπορεί κανείς να αναγνωρίσει και να μεταφέρει τα χαρακτηριστικά ενός συγκεκριμένου στυλ σε μια εικόνα διαφορετικού στυλ.

Σε αυτή τη διπλωματική εργασία, προσπαθούμε να κάνουμε τη διάκριση μεταξύ των Δυτικών και Ανατολικών στυλ κινούμενων σχεδίων (συχνά αναφέρονται ως “cartoon” και “anime” αντίστοιχα). Αυτό χωρίζεται σε δυο μέρη. Το πρώτο μέρος είναι η αναγνώριση και η ταξινόμηση στυλ, όπου προσπαθούμε να ταξινομήσουμε την κατηγορία μιας εικόνας και το δεύτερο μέρος είναι η μεταφορά στυλ, όπου προσπαθούμε να μεταφέρουμε τα χαρακτηριστικά του ενός στυλ σε μια εικόνα του άλλου.

Για να το πετύχουμε αυτό, δημιουργούμε και αξιολογούμε μερικά μοντέλα CNN, με διαφορετικές αρχιτεκτονικές και διαφορετικά επίπεδα πολυπλοκότητας στην αρχιτεκτονική τους, και συγκρίνουμε την απόδοση και τα αποτελέσματά τους. Πιο συγκεκριμένα, για το κομμάτι της αναγνώρισης και ταξινόμησης, δημιουργήσαμε 3 διαφορετικά μοντέλα, ένα VGG16 μοντέλο, ένα VGG19 μοντέλο και ένα μοντέλο CNN με 4 συνελκτικά επίπεδα, τα οποία εκπαιδεύσαμε, αξιολογήσαμε και συγκρίναμε τις επιδόσεις και τα αποτελέσματά τους. Για το κομμάτι της μεταφοράς στυλ, δημιουργήσαμε ένα VGG19 μοντέλο και χρησιμοποιώντας συγκεκριμένα στρώματα, τα οποία περιέχουν την πληροφορία που μας ενδιαφέρει, μεταφέραμε τα χαρακτηριστικά στυλ από μια εικόνα σε μια άλλη.

Λέξεις κλειδιά

Μηχανική Μάθηση, Βαθιά Μάθηση, Συνελκτικά Νευρωνικά Δίκτυα, CNN, Εντοπισμός, Αναγνώριση, Ταξινόμηση, Μεταφορά στυλ, VGG

Abstract

Image recognition and classification are the two most common use cases of CNNs in the field of Machine Learning. Going a step further, detecting and classifying different styles, as well as transferring characteristics of a style, from one image to another is an interesting application of CNNs.

The goal is to create a model which, given enough images of different styles (whether that would be different art styles, animation styles etc.), recognize, differentiate and accurately predict the style of an image. For the second part, using parts of a model, one can recognize and transfer the characteristics of a particular style to an image of a different style.

In this diploma thesis, we endeavor to differentiate between Western and Eastern animation styles (more commonly referred to as “cartoon” and “anime” respectively). This is split into two parts. The first part is style recognition and classification, where we try to classify the category of an image and the second part is style transfer, where we try to transfer the characteristics of each style to an image of the different style.

To achieve this, we create and evaluate a few CNN models, with different architectures and differing levels of complexity in their designs, and compare their performance and results. In particular, for the style recognition part, we create 3 different models, a VGG16 model, a VGG19 model and a custom CNN model with 4 convolution layers, which we train, evaluate and compare their performance and results. As for the style transfer part, we create a VGG19 model and using specific layers, which contain the information we need, we transfer the style characteristics from one image on to another.

Key Words

Machine Learning, Deep Learning, Convolutional Neural Network, CNN, Detection, Recognition, Classification, Style Transfer, VGG

Περιεχόμενα

Πίνακας Περιεχομένων

Περίληψη.....	6
Abstract.....	7
Περιεχόμενα.....	8
Κατάλογος σχημάτων.....	10
Κεφάλαιο 1 - Εισαγωγή.....	12
1.1 Περιγραφή προβλήματος.....	12
1.2 Συμβολή.....	12
1.3 Δομή εργασίας.....	13
Κεφάλαιο 2 - Συνελικτικά Νευρωνικά Δίκτυα.....	14
2.1 Επίπεδα.....	14
2.1.1 Συνελικτικό στρώμα.....	14
2.1.2 Συγκεντρωτικό στρώμα.....	15
2.1.3 Πλήρως Συνδεδεμένο στρώμα.....	15
2.1.4 Στρώμα εγκατάλειψης.....	15
2.2 Συναρτήσεις ενεργοποίησης.....	16
2.2.1 Γραμμική.....	16
2.2.2 Σιγμοειδής.....	16
2.2.3 Συνάρτηση υπερβολικής εφαπτομένης.....	17
2.2.4 Συνάρτηση διορθωμένης γραμμικής μονάδας.....	18
2.2.5 Softmax.....	19
2.3 Αρχιτεκτονική.....	19
2.3.1 Βασικό CNN.....	19
2.3.2 VGG16.....	20
2.3.3 VGG19.....	21
2.4 Επιλογή συνόλου δεδομένων.....	21
2.5 Μέθοδοι εκπαίδευσης.....	22

2.5.1 Μηχανική μάθηση.....	22
2.5.2 Μάθηση Μεταφοράς.....	22
2.6 Συχνά προβλήματα.....	23
2.6.1 Απαιτούμενοι πόροι.....	23
2.6.2 Υπερεκπαίδευση.....	23
2.7 Τεχνικές βελτιστοποίησης εκπαίδευσης.....	23
2.7.1 Χωρισμός συνόλου δεδομένων σε υποσύνολα.....	23
2.7.2 Κανονικοποίηση δέσμης δεδομένων.....	24
Κεφάλαιο 3 - Αναγνώριση, ταξινόμηση και μεταφορά στυλ.....	25
3.1 Αναγνώριση και Ταξινόμηση στυλ.....	25
3.2 Μεταφορά στυλ.....	26
Κεφάλαιο 4 - Πειραματική Αξιολόγηση.....	28
4.1 Αναγνώριση και ταξινόμηση στυλ.....	28
4.1.1 Σύνολο δεδομένων.....	28
4.1.2 Τρόποι Αξιολόγησης.....	29
4.1.3 Αρχιτεκτονικές μοντέλων.....	31
4.1.4 Αποτελέσματα.....	34
4.1.5 Σύγκριση αποτελεσμάτων.....	45
4.2 Μεταφορά στυλ.....	46
4.2.1 Σύνολο δεδομένων.....	46
4.2.2 Τρόποι αξιολόγησης.....	47
4.2.3 Αρχιτεκτονική μοντέλου.....	49
4.2.4 Αποτελέσματα.....	50
Κεφάλαιο 5 - Συμπεράσματα και Βελτιώσεις.....	57
Βιβλιογραφία.....	58
Παράρτημα.....	61
Ευρετήριο Ακρωνυμίων και Συντμήσεων.....	61

Κατάλογος σχημάτων

Σχήμα 1 : Γραμμική συνάρτηση ενεργοποίησης.....	16
Σχήμα 2 : Σιγμοειδής συνάρτηση ενεργοποίησης.....	17
Σχήμα 3 : Συνάρτηση υπερβολικής εφαπτομένης.....	18
Σχήμα 4 : Συνάρτηση διορθωμένης γραμμικής μονάδας.....	18
Σχήμα 5 : Παράδειγμα αρχιτεκτονικής CNN.....	20
Σχήμα 6 : Παράδειγμα αρχιτεκτονικής VGG16 μοντέλου.....	21
Σχήμα 7 : Παράδειγμα μεταφοράς στυλ.....	26
Σχήμα 8 : Παράδειγμα πίνακα σύγκυσης.....	31
Σχήμα 9 : Αρχιτεκτονική μοντέλου CNN4.....	32
Σχήμα 10 : Αρχιτεκτονική μοντέλου VGG16.....	33
Σχήμα 11 : Αποτελέσματα κώδικα ρυθον του μοντέλου cnn4.....	35
Σχήμα 12 : Διαγράμματα συνολικής ακρίβειας και απώλειας του μοντέλου cnn4.....	35
Σχήμα 13 : Πίνακας σύγκυσης (σύνολο εκπαίδευσης) του μοντέλου cnn4.....	36
Σχήμα 14 : Αποτελέσματα classifier του μοντέλου cnn4.....	36
Σχήμα 15 : Πίνακας σύγκυσης (σύνολο τεστ) του μοντέλου cnn4.....	37
Σχήμα 16 : Παραδείγματα σωστής αναγνώρισης του μοντέλου cnn4.....	37
Σχήμα 17 : Παραδείγματα λανθασμένης αναγνώρισης του μοντέλου cnn4.....	38
Σχήμα 18 : Αποτελέσματα κώδικα ρυθον του μοντέλου VGG16.....	38
Σχήμα 19 : Διαγράμματα συνολικής ακρίβειας και απώλειας του μοντέλου VGG16.....	39
Σχήμα 20 : Πίνακας σύγκυσης (σύνολο εκπαίδευσης) του μοντέλου VGG16.....	39
Σχήμα 21 : Αποτελέσματα classifier του μοντέλου VGG16.....	40
Σχήμα 22 : Πίνακας σύγκυσης (σύνολο τεστ) του μοντέλου VGG16.....	40
Σχήμα 23 : Παραδείγματα σωστής αναγνώρισης του μοντέλου VGG16.....	41
Σχήμα 24 : Παραδείγματα λανθασμένης αναγνώρισης του μοντέλου VGG16.....	41
Σχήμα 25 : Αποτελέσματα κώδικα ρυθον του μοντέλου VGG19.....	42
Σχήμα 26 : Διαγράμματα συνολικής ακρίβειας και απώλειας του μοντέλου VGG19.....	42
Σχήμα 27 : Πίνακας σύγκυσης (σύνολο εκπαίδευσης) του μοντέλου VGG19.....	43

Σχήμα 28 : Αποτελέσματα classifier του μοντέλου VGG19.....	43
Σχήμα 29 : Πίνακας σύγκρισης (σύνολο τεστ) του μοντέλου VGG19.....	44
Σχήμα 30 : Παραδείγματα σωστής αναγνώρισης του μοντέλου VGG19.....	44
Σχήμα 31 : Παραδείγματα λανθασμένης αναγνώρισης του μοντέλου VGG19.....	45
Σχήμα 32 : Διάγραμμα σύγκρισης αποτελεσμάτων στο σύνολο εκπαίδευσης.....	45
Σχήμα 33 : Διάγραμμα σύγκρισης αποτελεσμάτων στο σύνολο τεστ.....	46
Σχήμα 34 : Εικόνες που επιλέχθηκαν για μεταφορά στυλ (Πείραμα 1).....	47
Σχήμα 35 : Αρχιτεκτονική μοντέλου μεταφοράς στυλ.....	50
Σχήμα 36 : Η σταδιακή μεταμόρφωση της εικόνας περιεχομένου (Πείραμα 1).....	50
Σχήμα 37 : Αποτέλεσμα μεταφοράς στυλ (Πείραμα 1).....	50
Σχήμα 38 : Εικόνες που επιλέχθηκαν για μεταφορά στυλ (Πείραμα 2).....	51
Σχήμα 39 : Η σταδιακή μεταμόρφωση της εικόνας περιεχομένου (Πείραμα 2).....	51
Σχήμα 40 : Αποτέλεσμα μεταφοράς στυλ (Πείραμα 2).....	52
Σχήμα 41 : Η σταδιακή μεταμόρφωση της εικόνας περιεχομένου (Πείραμα 3).....	52
Σχήμα 42 : Αποτέλεσμα μεταφοράς στυλ (Πείραμα 3).....	53
Σχήμα 43 : Η σταδιακή μεταμόρφωση της εικόνας περιεχομένου (Πείραμα 4).....	53
Σχήμα 44 : Αποτέλεσμα μεταφοράς στυλ (Πείραμα 4).....	54
Σχήμα 45 : Διάγραμμα σύγκρισης αποτελεσμάτων μεταφοράς στυλ 1000 επαναλήψεων (Πειράματα 1 και 2).....	55
Σχήμα 46 : Διάγραμμα σύγκρισης αποτελεσμάτων μεταφοράς στυλ 10000 επαναλήψεων (Πειράματα 3 και 4).....	55
Σχήμα 47 : Συγκεντρωτικό διάγραμμα αποτελεσμάτων μεταφοράς στυλ.....	55

Κεφάλαιο 1 - Εισαγωγή

Η ιστορία των Νευρωνικών Δικτύων και της Μηχανικής μάθησης ξεκινά με την επιτυχή δημιουργία του πρώτου τεχνητού νευρώνα από τους Warren McCulloch και Walter Pitts το 1943 ([McCulloch and Pitts, 1943](#)). Η σπουδαία αυτή ανακάλυψη μπορούσε να εκτελέσει απλές λογικές πράξεις και ήταν το πρώτο βήμα στην δημιουργία του Perceptron, το οποίο είναι το πιο βασικό στοιχείο των Νευρωνικών Δικτύων (Neural Networks).

Η δημιουργία του Perceptron οφείλεται στον Frank Rosenblatt, ο οποίος, το 1957, συνέδεσε το μοντέλο αλληλεπίδρασης των εγκεφαλικών κυττάρων του Donald Hebb με τις γνώσεις και εμπειρία του Arthur Samuel στον τομέα του Machine Learning ([Rosenblatt, 1958](#)). Παρόλο που το Perceptron ήταν μια σπουδαία ανακάλυψη, είχε αρκετά προβλήματα, όπως την αδυναμία αναγνώρισης διαφόρων οπτικών προτύπων, τα οποία οδήγησαν σε καθυστερήσεις στην έρευνα των Νευρωνικών Δικτύων.

Μετά από δεκαετίες σπουδαίων ανακαλύψεων, όπως ο αλγόριθμος του κοντινότερου γείτονα ([Altman, 1992](#)) ή την χρήση Πολυστρωματικών Νευρωνικών Δικτύων, η Μηχανική Μάθηση είναι υπεύθυνη για ένα μεγάλο ποσοστό της ραγδαίας τεχνολογικής εξέλιξης και χρησιμοποιείται στους περισσότερους τομείς της κοινωνίας. ([Chandra, 2018](#); [Foote, 2019](#))

1.1 Περιγραφή προβλήματος

Υπάρχουν διάφορα προβλήματα Μηχανικής Μάθησης τα οποία απασχολούν την επιστημονική κοινότητα, καθώς και διάφορους προγραμματιστές που ασχολούνται με το αντικείμενο. Για την αντιμετώπιση αυτών των προβλημάτων, αλλά και για την βελτίωση υπαρχόντων συστημάτων, έχει δημιουργηθεί ένας σημαντικός αριθμός μοντέλων, τα οποία εκπαιδεύονται σε διάφορα σύνολα δεδομένων, ανάλογα με το είδος της εφαρμογής που θα έχουν.

Το αντικείμενο της συγκεκριμένης διπλωματικής εργασίας αφορά την μελέτη των προβλημάτων της αναγνώρισης στυλ, καθώς και της μεταφοράς στυλ, χρησιμοποιώντας μοντέλα Νευρωνικών Δικτύων τα οποία χρησιμοποιούνται συχνά σε έρευνες και την σύγκριση των αποτελεσμάτων τους με αυτά ενός μοντέλου που δημιουργήθηκε για αυτή την εργασία, η αρχιτεκτονική του οποίου είναι αρκετά πιο απλή.

1.2 Συμβολή

Όπως αναφέρθηκε νωρίτερα, το αντικείμενο αυτής της εργασίας είναι η ερευνητική ενασχόληση με τα προβλήματα της αναγνώρισης και μεταφοράς στυλ. Για την πραγματοποίηση αυτού του σκοπού πραγματοποιήθηκαν τα εξής:

- Παραγωγή πηγαίου κώδικα για την δημιουργία συνόλου δεδομένων που θα χρησιμοποιηθεί για την εκπαίδευση των μοντέλων.
- Παραγωγή πηγαίου κώδικα για την δημιουργία και εκπαίδευση ενός Συνελικτικού Νευρωνικού Δικτύου με 4 επίπεδα συνέλιξης.
- Παραγωγή πηγαίου κώδικα για την δημιουργία δύο μοντέλων, ένα VGG16 και ένα VGG19, και εκπαίδευσή τους με την τεχνική της Μεταφοράς Μάθησης.
- Σύγκριση των αποτελεσμάτων των 3 αυτών μοντέλων.
- Παραγωγή πηγαίου κώδικα για την μεταφορά στυλ μεταξύ εικόνων
- Αξιολόγηση των αποτελεσμάτων μεταφοράς στυλ.

1.3 Δομή εργασίας

Η εργασία αυτή είναι χωρισμένη σε 6 κεφάλαια. Πέρα από την εισαγωγή (*Κεφάλαιο 1 - Εισαγωγή*), στο κεφάλαιο 2 (*Κεφάλαιο 2 - Συνελικτικά Νευρωνικά Δίκτυα*) καλύπτεται το θεωρητικό υπόβαθρο των Συνελικτικών Νευρωνικών Δικτύων για να είναι δυνατή η κατανόηση της αρχιτεκτονικής των μοντέλων που παρουσιάζονται σε επόμενα κεφάλαια. Στο κεφάλαιο 3 (*Κεφάλαιο 3 - Αναγνώριση, ταξινόμηση και μεταφορά στυλ*) γίνεται μια σύντομη περιγραφή των προβλημάτων της αναγνώρισης και ταξινόμησης στυλ, καθώς και του προβλήματος της μεταφοράς στυλ. Στο κεφάλαιο 4 (*Κεφάλαιο 4 - Πειραματική Αξιολόγηση*) παρουσιάζονται και αναλύονται τα πειράματα που πραγματοποιήθηκαν για αυτή την εργασία, καθώς και τα αποτελέσματα που προέκυψαν από τα πειράματα αυτά. Τέλος, στο κεφάλαιο 5 (*Κεφάλαιο 5 - Συμπεράσματα και Βελτιώσεις*) ολοκληρώνεται η εργασία, παρουσιάζοντας τα γενικά συμπεράσματα που προέκυψαν από την έρευνα και τον πειραματισμό με τα θέματα της αναγνώρισης, ταξινόμησης και μεταφοράς στυλ, καθώς και πιθανές βελτιώσεις που θα μπορούσαν να γίνουν στο μέλλον, για την επίτευξη ακόμα καλύτερων αποτελεσμάτων.

Κεφάλαιο 2 - Συνελικτικά Νευρωνικά Δίκτυα

Τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks - CNNs) είναι μία υποκατηγορία των Νευρωνικών Δικτύων, τα οποία χρησιμοποιούνται κυρίως για την ανάλυση εικόνων. Ονομάζονται έτσι καθώς τουλάχιστον ένα από τα επίπεδά (στρώματα) τους χρησιμοποιούν την πράξη της συνέλιξης. ([Valueva et al., 1990](#); [Zhang, Itoh, Tanida and Ichioka, 1990](#))

2.1 Επίπεδα

Κάθε μοντέλο αποτελείται από ένα στρώμα εισόδου (Input Layer), μέσω του οποίου εισέρχονται τα δεδομένα εισόδου (Input Data) από το σύνολο δεδομένων (Dataset) και ένα στρώμα εξόδου (Output Layer), το οποίο εξάγει τα τελικά αποτελέσματα (Output Data) του δικτύου. Εκτός από τα στρώματα αυτά, υπάρχουν και άλλα, τα οποία ονομάζονται κρυφά ή ενδιάμεσα στρώματα (Hidden Layers), μερικά εκ των οποίων παρουσιάζονται παρακάτω. ([MK Gurucharan, 2020](#))

2.1.1 Συνελικτικό στρώμα

Το συνελικτικό στρώμα (Convolution Layer) είναι το υπεύθυνο για την εξαγωγή των διαφόρων χαρακτηριστικών από τις εικόνες εισόδου. Εδώ γίνεται η συνέλιξη μεταξύ μια εικόνας εισόδου και ενός σετ φίλτρων (Πυρήνες - Kernels) συγκεκριμένου μεγέθους (μικρότερου από αυτό της εισόδου) και έτσι παράγεται η έξοδος, η οποία ονομάζεται χάρτης χαρακτηριστικών (Feature Map).

Ο χάρτης χαρακτηριστικών μας δίνει κάποιες πληροφορίες σχετικά με την εικόνα εισόδου, όπως είναι οι γωνίες και οι ακμές (Corners and Edges). Αυτός ο χάρτης χρησιμοποιείται και ως είσοδος σε άλλα στρώματα, τα οποία μπορούν να μάθουν ακόμα περισσότερα χαρακτηριστικά της εικόνας εισόδου.

Το μέγεθος του χάρτη χαρακτηριστικών προκύπτει από την διάσταση εισόδου W , το μέγεθος του πυρήνα K των νευρώνων του στρώματος, του βήματος S και την ποσότητα μηδενικού γεμίματος περιγράμματος (Zero Padding) P . Ο αριθμός των νευρώνων που ταιριάζουν σε έναν δεδομένο όγκο δίνεται από την εξίσωση:

$$\frac{W - K + 2P}{S} + 1$$

2.1.2 Συγκεντρωτικό στρώμα

Το συγκεντρωτικό στρώμα (Pooling Layer) συνήθως ακολουθεί ένα συνελικτικό στρώμα. Ο σκοπός αυτό του στρώματος είναι η μείωση του μεγέθους του χάρτη χαρακτηριστικών, ώστε να επιτευχθεί μείωση του υπολογιστικού κόστους. Αυτό πραγματοποιείται μειώνοντας τις συνδέσεις μεταξύ των στρωμάτων και λειτουργεί ανεξάρτητα σε κάποια υποπεριοχή κάθε χάρτη χαρακτηριστικών, με αποτέλεσμα αυτός να υποδειγματοληπτείται.

Οι τρεις συνηθέστεροι μέθοδοι συγκέντρωσης είναι:

- Συγκέντρωση Μέγιστου (Max Pooling): Η τελική τιμή είναι η τιμή του μεγαλύτερου στοιχείου της υποπεριοχής του χάρτη χαρακτηριστικών.
- Συγκέντρωση Ελάχιστου (Min Pooling): Η τελική τιμή είναι η τιμή του μικρότερου στοιχείου της υποπεριοχής του χάρτη χαρακτηριστικών.
- Συγκέντρωση Μέσης Τιμής (Average Pooling): Η τελική τιμή είναι ο μέσος όρος όλων των τιμών της υποπεριοχής του χάρτη χαρακτηριστικών.

2.1.3 Πλήρως Συνδεδεμένο στρώμα

Το πλήρως συνδεδεμένο στρώμα (Fully Connected Layer - FC) πέρα από νευρώνες αποτελείται και από τα βάρη και τις προκαταλήψεις (Weights and Biases) και χρησιμοποιείται για τη σύνδεση των νευρώνων διαφορετικών στρωμάτων. Γι' αυτό και κάθε νευρώνας αυτού του στρώματος είναι συνδεδεμένος με κάθε νευρώνα του προηγούμενου στρώματος. Αυτό το στρώμα συνήθως τοποθετείται πριν το στρώμα εξόδου. Σε αυτό το στάδιο γίνεται η ταξινόμηση των δειγμάτων σε διάφορες κλάσεις.

2.1.4 Στρώμα εγκατάλειψης

Συχνά, όταν όλα τα χαρακτηριστικά είναι συνδεδεμένα με το πλήρως συνδεδεμένο στρώμα, υπάρχει μεγάλη πιθανότητα να δημιουργηθεί το πρόβλημα της υπερεκπαίδευσης (Overfitting) στο σύνολο των δεδομένων εκπαίδευσης. Το πρόβλημα της υπερεκπαίδευσης θα μελετηθεί αργότερα σε αυτήν την εργασία.

Για να ξεπεραστεί αυτό το πρόβλημα, χρησιμοποιείται ένα στρώμα εγκατάλειψης (Dropout Layer) στο οποίο μερικές συνδέσεις μεταξύ νευρώνων του δικτύου δεν χρησιμοποιούνται κατά την διάρκεια της εκπαίδευσης, με αποτέλεσμα να μειώνεται το μέγεθος του μοντέλου. Αν για παράδειγμα είχαμε ένα τέτοιο στρώμα με dropout 0.3, το 30% των νευρώνων, με τυχαία επιλογή, "απομακρύνονται" από το νευρωνικό δίκτυο.

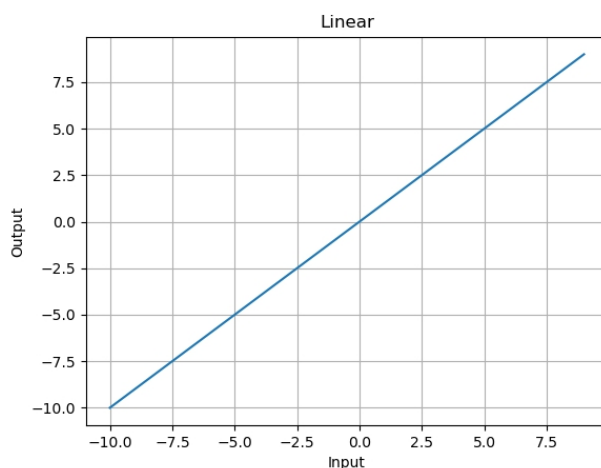
2.2 Συναρτήσεις ενεργοποίησης

Μία από τις πιο σημαντικές παραμέτρους ενός CNN μοντέλου είναι η συνάρτηση ενεργοποίησης (Activation Function). Οι συναρτήσεις ενεργοποίησης χρησιμοποιούνται ώστε ένα μοντέλο να μπορεί να μάθει και να προσεγγίσει τις σχέσεις που υπάρχουν μεταξύ των μεταβλητών του δικτύου, εισάγοντας τον παράγοντα της μη γραμμικότητας (non-linearity). Με άλλα λόγια, αποφασίζει ποια πληροφορία του μοντέλου πρέπει να προχωρήσει στο επόμενο στρώμα.

Υπάρχουν διάφορες συναρτήσεις ενεργοποίησης, όπως είναι η ReLU, η Softmax, η tanH και η Sigmoid. Κάθε μια από αυτές τις συναρτήσεις έχει συγκεκριμένη χρήση. Για παράδειγμα ένα μοντέλο δυαδικής ταξινόμησης (Binary Classification) συνήθως χρησιμοποιεί Sigmoid, ενώ για ταξινόμηση πολλαπλών κλάσεων (Multi-class Classification) προτιμάται η Softmax. Ακόμα και στο ίδιο μοντέλο μπορεί υπάρχουν διάφορες συναρτήσεις συμμετοχής σε διαφορετικά στρώματα. ([Brownlee, 2021](#))

2.2.1 Γραμμική

Η γραμμική (Linear) συνάρτηση είναι η πιο απλή συνάρτηση ενεργοποίησης, καθώς η έξοδος της ισούται με την είσοδο, χωρίς να γίνει καμία αλλαγή. Γι' αυτόν τον λόγο η γραφική της παράσταση είναι μια διαγώνια γραμμή όπου κάθε σημείο της έχει τετμημένη ίση με την τεταγμένη (π.χ. (0, 0) ή (5, 5)) όπως φαίνεται παρακάτω:



Σχήμα 1: Γραμμική συνάρτηση ενεργοποίησης

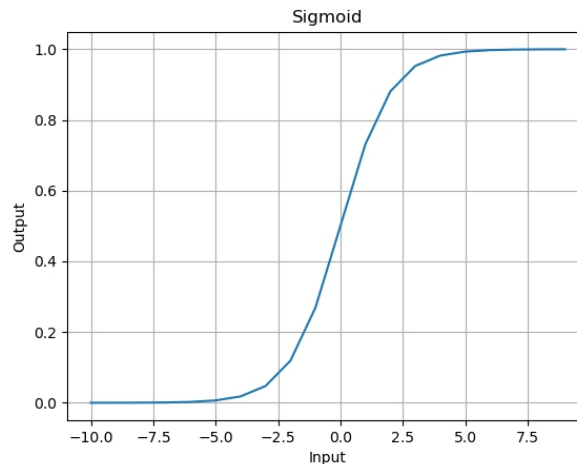
2.2.2 Σιγμοειδής

Η σιγμοειδής (Sigmoid) συνάρτηση δέχεται ως είσοδο οποιαδήποτε πραγματική τιμή και παράγει ως έξοδο τιμές μεταξύ 0 και 1. Όσο μεγαλύτερη είναι η είσοδος, τόσο πιο κοντά στο 1 θα

είναι ή έξοδος και αντίστροφα, όσο πιο μικρή είναι η είσοδος, τόσο πιο κοντά στο 0 θα είναι η έξοδος. Ορίζεται από την συνάρτηση:

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

Το όνομα της συνάρτησης οφείλεται στην της, η οποία έχει την χαρακτηριστική μορφή “S”, όπως φαίνεται παρακάτω:



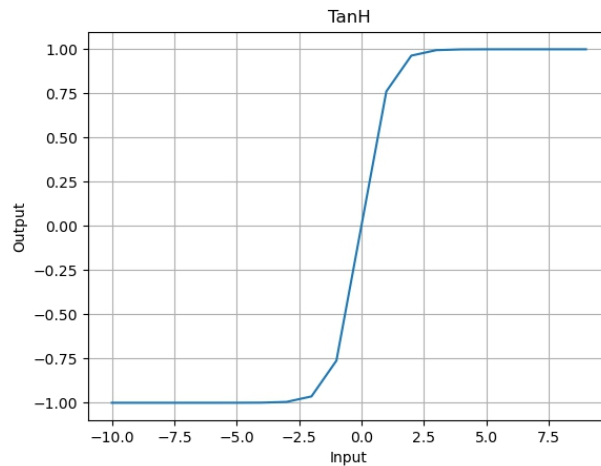
Σχήμα 2: Σιγμοειδής συνάρτηση ενεργοποίησης

2.2.3 Συνάρτηση υπερβολικής εφαπτομένης

Η συνάρτηση υπερβολικής εφαπτομένης (Hyperbolic Tangent ή tanh) μοιάζει αρκετά με την σιγμοειδή, με την διαφορά ότι οι τιμές εξόδου κυμαίνονται μεταξύ -1 και 1. Όσο μεγαλύτερη είναι η είσοδος, τόσο πιο κοντά στο ένα θα είναι η έξοδος και αντίστροφα, όσο πιο μικρή είναι η είσοδος, τόσο πιο κοντά στο -1 θα είναι η έξοδος. Ορίζεται από την συνάρτηση:

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

Το σχήμα της συνάρτησης αυτής μοιάζει με αυτό της σιγμοειδής, όπως φαίνεται παρακάτω:



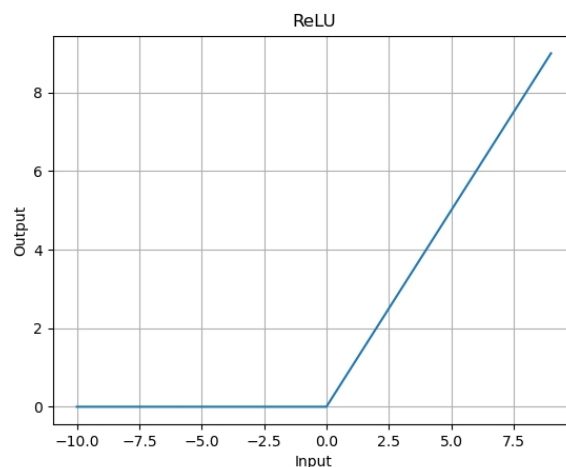
Σχήμα 3: Συνάρτηση υπερβολικής εφαπτομένης

2.2.4 Συνάρτηση διορθωμένης γραμμικής μονάδας

Η συνάρτηση διορθωμένης γραμμικής μονάδας (ReLU) είναι ίσως η συνάρτηση που χρησιμοποιείται πιο συχνά σε ενδιάμεσα στρώματα. Αυτό συμβαίνει γιατί είναι αρκετά εύκολο να υλοποιηθεί και γιατί έχει παρατηρηθεί πως η χρήση της μικραίνει τον κύκλο εκπαίδευσης. Η συνάρτηση αυτή είναι λιγότερο ευαίσθητη στο πρόβλημα εξαφάνισης κλίσεων (Vanishing Gradients problem), το οποίο αντιμετωπίζουν οι προηγούμενες συναρτήσεις ενεργοποίησης, αλλά μπορεί να συναντήσει άλλα προβλήματα κορεσμένων ή “νεκρών” μονάδων, όπου δίνει την ίδια έξοδο για οποιαδήποτε είσοδο. Ορίζεται από την συνάρτηση:

$$\text{relu}(x) = \max(0, x)$$

Αυτό σημαίνει ότι για αρνητικές τιμές εισόδου, η έξοδος είναι πάντα 0, ενώ για θετικές τιμές εισόδου, είναι στην ουσία σαν την γραμμική συνάρτηση, όπως φαίνεται παρακάτω:



Σχήμα 4: Συνάρτηση διορθωμένης γραμμικής μονάδας

2.2.5 Softmax

Η συνάρτηση ενεργοποίησης softmax διαφέρει από της προηγούμενες συναρτήσεις καθώς αντί για μια τιμή εισόδου, δέχεται ένα διάνυσμα πραγματικών τιμών και παράγει ως έξοδο ένα διάνυσμα τιμών που ανέρχονται στο 1, οι οποίες ερμηνεύονται ως πιθανότητες συμμετοχής για την κάθε κλάση. Ορίζεται από την συνάρτηση:

$$\text{softmax}(x) = \frac{e^x}{\sum e^x}$$

2.3 Αρχιτεκτονική

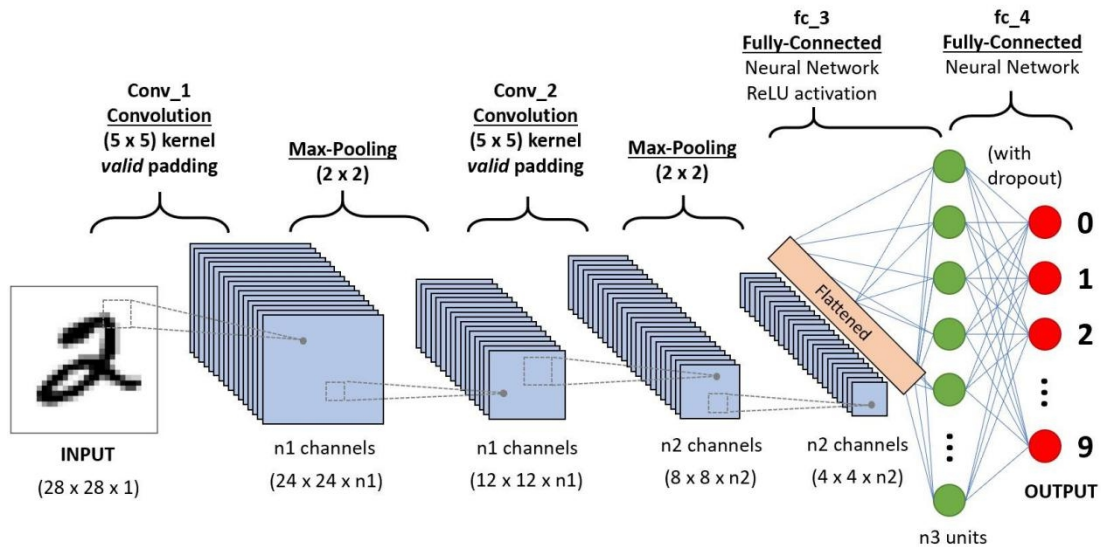
Κάθε CNN αποτελείται από δύο βασικά κομμάτια:

- Ένα συνελικτικό επίπεδο, το οποίο αναγνωρίζει και χωρίζει διάφορα χαρακτηριστικά μίας εικόνας για ανάλυση, η οποία ονομάζεται Εξαγωγή Χαρακτηριστικών (Feature Extraction)
- Ένα πλήρως συνδεδεμένο επίπεδο, το οποίο χρησιμοποιεί την έξοδο της συνέλιξης και προβλέπει την κλάση της εικόνας με βάση τα χαρακτηριστικά που έχουν εξαχθεί σε προηγούμενο στάδιο.

Τα επίπεδα αυτά χρησιμοποιούνται σε συνδυασμό με άλλα επίπεδα και συναρτήσεις συμμετοχής, όπως αυτά που αναφέρθηκαν νωρίτερα, για την δημιουργία και την βελτίωση ενός μοντέλου .

2.3.1 Βασικό CNN

Ένα απλό CNN αποτελείται από ένα επίπεδο εισόδου, έναν αριθμό ενδιάμεσων, κρυφών επιπέδων και ένα επίπεδο εξόδου. Τα ενδιάμεσα μοντέλα ονομάζονται και κρυφά καθώς οι εισοδοι και οι έξοδοι τους “κρύβονται” πίσω από κάποια συνάρτηση ενεργοποίησης. Παρακάτω παρουσιάζεται μία απλή αρχιτεκτονική CNN, η οποία έχει 2 επίπεδα συνέλιξης:



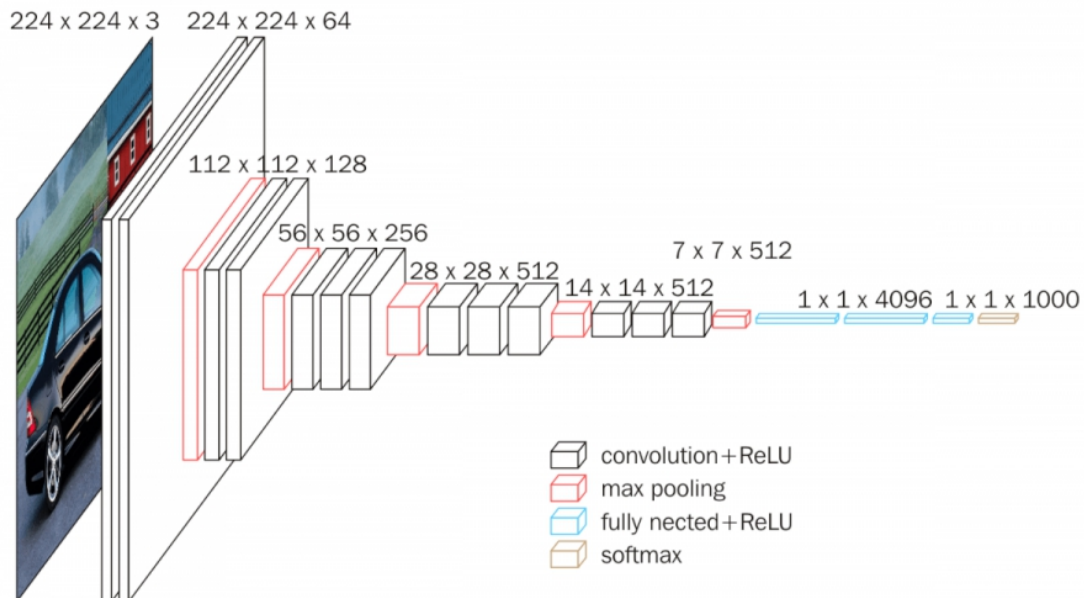
Σχήμα 5: Παράδειγμα αρχιτεκτονικής CNN.

Στο παράδειγμα αυτό βλέπουμε το στρώμα εισόδου στα αριστερά, τα ενδιάμεσα στρώματα και το στρώμα εξόδου στα δεξιά, το οποίο επιλέγει μία από τις δέκα πιθανές κλάσεις ως αποτέλεσμα.

(Πηγή εικόνας: <https://www.analyticsvidhya.com/blog/2020/10/what-is-the-convolutional-neural-network-architecture/>) (PHANI8, 2020)

2.3.2 VGG16

Το VGG16 είναι μια εκδοχή του VGG μοντέλου που προτάθηκε από τους Simonyan και Zisserman (Simonyan and Zisserman, 2014). Τα VGG μοντέλα χρησιμοποιούνται πολύ συχνά σε έρευνες που αφορούν τη Μηχανική Μάθηση, κυρίως σε προβλήματα βαθιάς μάθησης σχετικά με ταξινόμηση εικόνας. Το 16 στο όνομά του προκύπτει από το γεγονός ότι έχει 13 στρώματα συνέλιξης και 3 πλήρως συνδεδεμένα στρώματα. Η αρχιτεκτονική του παρουσιάζεται παρακάτω:



Σχήμα 6: Παράδειγμα αρχιτεκτονικής VGG16 μοντέλου

Στο παράδειγμα αυτό βλέπουμε πως το μοντέλο αυτό δέχεται μια RGB εικόνα μεγέθους 224 x 224 και όλα τα ενδιάμεσα στρώματα είναι εξοπλισμένα με την συνάρτηση διορθωμένης γραμμικότητας (ReLU). (Πηγή εικόνας: <https://neurohive.io/en/popular-networks/vgg16/>) ([VGG16 - Convolutional Network for Classification and Detection, 2011](#))

2.3.3 VGG19

Το VGG19 είναι μια άλλη παραλλαγή του VGG μοντέλου και η αρχιτεκτονική του είναι παρόμοια με αυτή του VGG16. Το 19 στο όνομά του προκύπτει από το γεγονός ότι έχει 16 στρώματα συνέλιξης (σε αντίθεση με τα 13 του VGG16) και 3 πλήρως συνδεδεμένα στρώματα.

2.4 Επιλογή συνόλου δεδομένων

Ένας από τους πιο σημαντικούς παράγοντες για την επίτευξη καλών αποτελεσμάτων στην εκπαίδευση ενός μοντέλου είναι η σωστή επιλογή του συνόλου δεδομένων. Μεταξύ άλλων, ένα σύνολο δεδομένων πρέπει να είναι αρκετά μεγάλο, ώστε να παρέχει όλες τις σημαντικές πληροφορίες που θα εξαχθούν ως περιγραφικά χαρακτηριστικά στα οποία θα δώσει έμφαση το μοντέλο. Επίσης πρέπει να περιλαμβάνει ένα σχετικά ίσο πλήθος από την κάθε κλάση του προβλήματος, αλλιώς το μοντέλο θα αναγνωρίζει και θα έχει τάση προς μια από τις κλάσεις.

2.5 Μέθοδοι εκπαίδευσης

Υπάρχουν διάφορες μέθοδοι εκπαίδευσης ενός μοντέλου μηχανικής μάθησης. Η επιλογή της πιο κατάλληλης μεθόδου εξαρτάται από πολλούς παράγοντες όπως είναι το είδος της εφαρμογής ή του προβλήματος προς επίλυση, ή οι διαθέσιμοι πόροι, τόσο σε υπολογιστική ικανότητα όσο και σε θέμα χρόνου. Παρακάτω παρουσιάζονται συνοπτικά οι μέθοδοι που χρησιμοποιήθηκαν σε αυτή την εργασία.

2.5.1 Μηχανική μάθηση

Η μηχανική μάθηση (Machine Learning) είναι η διαδικασία με την οποία ένα δίκτυο μπορεί να καταλάβει και να μάθει την δομή ενός συνόλου δεδομένων και στην συνέχεια να μπορεί να πάρει κάποιες αποφάσεις σε κάποιο άλλο σχετικό σύνολο με καινούργια δεδομένα. Πέρα από αρκετό χρόνο, οι τεχνικές μηχανικής μάθησης απαιτούν αρκετά μεγάλο όγκο δεδομένων για να έχουν καλή απόδοση, καθώς και πολύ υπολογιστική ισχύ. ([Advani, 2021](#); [Machine Learning: What it is and why it matters, 2021](#))

2.5.2 Μάθηση Μεταφοράς

Η μάθηση μεταφοράς (Transfer Learning) είναι μια τεχνική μηχανικής μάθησης όπου ένα μοντέλο το οποίο έχει αναπτυχθεί και εκπαιδευτεί για κάποια εργασία χρησιμοποιείται ως σημείο έναρξης για χρήση σε κάποια άλλη σχετική εργασία. Χρησιμοποιείται αρκετά συχνά σε εφαρμογές όρασης υπολογιστών (Computer Vision) ή επεξεργασίας φυσικής γλώσσας (NLP - Natural Language Processing) καθώς τέτοιες εφαρμογές απαιτούν μεγάλη ποσότητα υπολογιστικών πόρων. ([Brownlee, 2017](#); [Castañón, 2019](#))

Ένα παράδειγμα εφαρμογής αυτής της τεχνικής είναι η εκπαίδευση ενός μοντέλου για την αναγνώριση αυτοκινήτων, και στην συνέχεια η χρήση του ίδιου μοντέλου για την αναγνώριση κάποιου άλλου είδους οχήματος, όπως είναι τα φορτηγά ή οι μηχανές. Για να επιτευχθεί αυτό πρέπει το μοντέλο να έχει μάθει κάποια γενικά χαρακτηριστικά από την πρώτη εφαρμογή, ώστε αφού εκπαιδευθεί με τα νέα δεδομένα, να μπορεί να αναγνωρίσει αυτά τα παρόμοια γενικά χαρακτηριστικά και στην δεύτερη εφαρμογή.

Συνεπώς, μετά από αρκετή έρευνα στο πόσο αποδοτική ή πρακτική είναι αυτή η μέθοδος στην εφαρμογή της, έχει επικρατήσει η τάση να δημιουργούνται διάφορα μοντέλα βαθιάς μάθησης, τα οποία είναι πολύ σύνθετα και εκπαιδούνται σε τεράστια σύνολα δεδομένων και στην συνέχεια δημοσιεύονται ώστε να μπορούν να χρησιμοποιηθούν σε διάφορες εργασίες ή έρευνες στο μέλλον. Ένα τέτοιο μοντέλο είναι και το VGG, το οποίο χρησιμοποιήθηκε σε αυτήν την εργασία. ([Pan and Yang, 2010](#); [Yosinski, Clune, Bengio and Lipson, 2014](#))

2.6 Συχνά προβλήματα

2.6.1 Απαιτούμενοι πόροι

Το πιο σημαντικό πρόβλημα που αντιμετωπίζουν τα νευρωνικά δίκτυα είναι οι πόροι που απαιτούνται για την εκπαίδευσή τους. Ανάλογα με τον όγκο των δεδομένων που θα χρησιμοποιηθούν για την εκπαίδευση ενός μοντέλου και το είδος του προβλήματος το οποίο καλείται να επιλύσει το μοντέλο θα χρειαστούν από λίγες ώρες μέχρι και βδομάδες ή μήνες για να ολοκληρωθεί η εκπαίδευση. Αυτό εξαρτάται φυσικά και από τους διαθέσιμους υπολογιστικούς πόρους, που σημαίνει πως για την γρηγορότερη ολοκλήρωση της διαδικασίας της εκπαίδευσης απαιτούνται ισχυροί υπολογιστές, συγκεκριμένα με ισχυρές κάρτες γραφικών. Εννοείται πως όσο μεγαλώνει η κλίμακα του έργου, τόσο αυξάνονται και οι απαιτήσεις των υπολογιστικών πόρων.

Αυτό είναι και το σημαντικότερο πρόβλημα που αντιμετώπισε αυτή η εργασία, καθώς οι υπολογιστικοί πόροι που ήταν διαθέσιμοι δεν ήταν αρκετοί για να αντιμετωπίσουν και το επόμενο πρόβλημα που περιγράφεται σε λογικά χρονικά πλαίσια.

2.6.2 Υπερεκπαίδευση

Η υπερεκπαίδευση (Overfitting) συμβαίνει όταν κάποιο μοντέλο λειτουργεί τόσο καλά στα δεδομένα εισόδου που προκαλεί μείωση της απόδοσης του μοντέλου σε νέα δεδομένα. Αυτό συμβαίνει γιατί το σύνολο δεδομένων δεν είναι χαρακτηριστικό του προβλήματος και το μοντέλο έμαθε να αναγνωρίζει κάποια χαρακτηριστικά που εμφανίζονται πάρα πολύ συχνά στο σύνολο εκπαίδευσης, αλλά δεν περιγράφουν ή δεν εμφανίζονται απαραίτητα στο σύνολο εκπαίδευσης.

Για την αντιμετώπιση αυτού το προβλήματος υπάρχουν διάφορες τεχνικές, μερικές από τις οποίες παρουσιάζονται παρακάτω.

2.7 Τεχνικές βελτιστοποίησης εκπαίδευσης

2.7.1 Χωρισμός συνόλου δεδομένων σε υποσύνολα

Όπως αναφέρθηκε και νωρίτερα, για την επιτυχή εκπαίδευση και αξιολόγηση ενός μοντέλου χρειάζεται ένα καλό σύνολο δεδομένων. Πιο συγκεκριμένα χρειάζεται ένα σύνολο εκπαίδευσης (Training set), πάνω στο οποίο θα εκπαιδευτεί το μοντέλο και μέσω αυτού θα αναγνωρίσει τα περιγραφικά χαρακτηριστικά, και ένα σύνολο τεστ (Test set), το οποίο θα χρησιμοποιηθεί για την αξιολόγηση των αποτελεσμάτων του μοντέλου. Είναι σημαντικό να αναφερθεί πως τα σύνολα αυτά δεν πρέπει να έχουν κοινά στοιχεία, καθώς αυτό επηρεάζει τα αποτελέσματα.

Ανάλογα το μοντέλο που χρησιμοποιείται, μπορεί να είναι απαραίτητη και η χρήση ενός ακόμα συνόλου, του συνόλου επικύρωσης (Validation set), το οποίο χρησιμοποιείται για την καλύτερη ρύθμιση των διάφορων παραμέτρων ενός μοντέλου.

Συνοπτικά λοιπόν έχουμε τα παρακάτω σύνολα:

- Σύνολο εκπαίδευσης: Το σύνολο που χρησιμοποιείται για την εκπαίδευση ενός μοντέλου.
- Σύνολο επικύρωσης: Το σύνολο που χρησιμοποιείται για την βελτίωση των παραμέτρων ενός μοντέλου.
- Σύνολο τεστ: Το σύνολο που χρησιμοποιείται για την αξιολόγηση του μοντέλου.

Ο χωρισμός και τα τελικά ποσοστά αυτών των συνόλων εξαρτάται από την εφαρμογή ή το πρόβλημα που καλείται να αντιμετωπίσει ένα μοντέλο. Στην εργασία αυτή, χωρίσαμε αρχικά το σύνολο δεδομένων σε δυο υποσύνολα, το σύνολο εκπαίδευσης και το σύνολο τεστ. Στην συνέχεια χωρίσαμε το σύνολο εκπαίδευσης σε δυο υποσύνολα, το πραγματικό σύνολο εκπαίδευσης και στο σύνολο επικύρωσης. Η τεχνική αυτή είναι γνωστή ως διασταυρούμενη επικύρωση (Cross Validation).

2.7.2 Κανονικοποίηση δέσμης δεδομένων

Η κανονικοποίηση δέσμης δεδομένων (Batch Normalization) προτάθηκε από τους Sergey Ioffe και Christian Szegedy ([Ioffe and Szegedy, 2015](#)) για την καταπολέμηση προβλημάτων αστάθειας κατά την διάρκεια της εκπαίδευσης ενός μοντέλου. Τα προβλήματα αυτά προκαλούνται όταν γίνεται αλλαγή στην κατανομή των δεδομένων, καθώς περνάνε από ένα επίπεδο σε ένα άλλο. Γίνεται κανονικοποίηση σε κάθε δέσμη δεδομένων, προσθέτοντας παραμέτρους κανονικοποίησης κατά την οπίσθια διάδοση σφάλματος. Έχει αποδειχθεί πως η χρήση της οδηγεί σε μικρότερους χρόνους σύγκλισης για νευρωνικά δίκτυα, πετυχαίνοντας και μεγαλύτερη ακρίβεια.

Κεφάλαιο 3 - Αναγνώριση, ταξινόμηση και

μεταφορά στυλ

3.1 Αναγνώριση και Ταξινόμηση στυλ

Η αναγνώριση και ταξινόμηση στυλ (Style Recognition and Classification) είναι η διαδικασία με την οποία εξάγονται διάφορα περιγραφικά χαρακτηριστικά από εικόνες με σκοπό να αναγνωριστούν το μοτίβο ή στυλ που έχουν κοινό οι εικόνες αυτές. Έχει εφαρμογή σε διάφορους τομείς όπως είναι η μόδα, η αρχιτεκτονική και η ζωγραφική. ([Lee and Cha, 2016](#); [Jiang et al., 2019](#))

Το μεγαλύτερο κομμάτι έρευνας έχει γίνει στον τομέα της ζωγραφικής, σχετικά με τον χαρακτηρισμό έργων τέχνης με την χρήση αλγορίθμων μηχανικής μάθησης σε ψηφιοποιημένα έργα τέχνης, δίνοντας έμφαση στην εξαγωγή διαφόρων χαρακτηριστικών όπως είναι το χρώμα, ο ζωγράφος, οι πινελιές ή η χρονική περίοδος. ([Rigau, Feixas and Sbert, 2008](#); [Vieira et al., 2015](#))

Οι πινελιές είναι ένα σημαντικό χαρακτηριστικό που μπορεί να που δηλώνει τον χαρακτήρα ενός ζωγράφου και μπορεί να χρησιμοποιηθεί για την αναγνώριση στυλ. Μπορούν να εξαχθούν με τον συνδυασμό των τεχνικών ανίχνευσης αιχμών (Edge Detection) και τμηματοποίησης βάσει συστάδων (Clustering-based Segmentation), με αποτέλεσμα διάφορες τιμές χαρακτηριστικών όπως είναι η τυπική απόκλιση του αριθμού ή του προσανατολισμού για πινελιές σε μια περιοχή, το μέγεθος, το μήκος ή η ομογένεια μπορούν να υπολογιστούν στατιστικά. Συγκεκριμένα για τον van Gogh έχει γίνει ανάλυση των έντονα ρυθμικών πινελιών του, τέτοια ανάλυση όμως είναι αρκετά περιορισμένη, καθώς μπορεί να γίνει για λίγους ζωγράφους που είναι πολύ μοναδικοί. ([Li, Yao, Hendriks and Wang, 2012](#); [Berezhnoy, Postma and van den Herik, 2007](#); [Johnson et al., 2008](#))

Σχετικά με την ταξινόμηση της τέχνης ανά στυλ, έχει γίνει έρευνα στην εξαγωγή χαρακτηριστικών ανά φως, γραμμή και χρώμα και ταξινόμηση των πινάκων τέχνης με την μέθοδο του πλησιέστερου γείτονα (k-nearest neighbor). Η οπτικοποίηση και ανάλυση των αποτελεσμάτων έγινε μέσω μη εποπτευόμενης μάθησης (Unsupervised Learning), ιεραρχικής ομαδοποίησης (Hierarchical Clustering) και αυτο-οργανωμένων χαρτών (Self-organizing maps). ([Lombardi, 2005](#))

Στην εργασία αυτή θα επικεντρωθούμε στον τομέα της σχεδίασης κινουμένων σχεδίων και στις διαφορές που παρουσιάζονται στο στυλ του ανατολικού κόσμου (“cartoon”) σε σχέση με το στυλ του δυτικού κόσμου (“anime”). Θα χρησιμοποιήσουμε βαθιά συνελκτικά νευρωνικά δίκτυα και τεχνικές μηχανικής μάθησης για την εξαγωγή περιγραφικών χαρακτηριστικών που θα είναι απαραίτητα για την διαφοροποίηση και αναγνώριση των δυο στυλ.

3.2 Μεταφορά στυλ

Μεταφορά στυλ (Neural Style Transfer - NST) είναι η τεχνική όρασης υπολογιστών (Computer Vision) που μας επιτρέπει να επανασυνθέσουμε το περιεχόμενο μιας εικόνας στο στυλ μιας άλλης. Με άλλα λόγια, χρησιμοποιώντας βαθιά μάθηση, μπορούμε να πάρουμε μια εικόνα και να την μετατρέψουμε στο στυλ κάποιας άλλης εικόνας. Για να επιτευχθεί αυτό χρειάζονται δύο εικόνες, μια εικόνα περιεχομένου (Content Image) και μια εικόνα στυλ (Style Reference Image), οι οποίες συνδυάζονται, με αποτέλεσμα η εικόνα εξόδου να είναι σαν την εικόνα περιεχομένου, αλλά “ζωγραφισμένη” με το στυλ της εικόνας στυλ. (Gatys, Ecker and Bethge, 2015; Johnson, 2019)



Σχήμα 7: Παράδειγμα μεταφοράς στυλ

Στο παράδειγμα αυτό βλέπουμε πως μπορούν να συνδυαστούν δύο εικόνες για να παραχθεί μια καινούργια (δεξιά εικόνα), η οποία μοιάζει με την αριστερή (εικόνα περιεχομένου), αλλά στο στυλ της μεσαίας εικόνας (εικόνα στυλ). (Πηγή εικόνας:

https://www.researchgate.net/publication/330828467_Towards_the_Algorithmic_Detection_of_Artistic_Style)

Η μεταφορά στυλ είναι ένα παράδειγμα σχηματοποίησης εικόνας (), η οποία είναι μια τεχνική επεξεργασίας εικόνας που μελετάται εδώ και αρκετές δεκαετίες.

Αν προσπαθήσουμε να λύσουμε το πρόβλημα αυτό χρησιμοποιώντας κλασικές τεχνικές εποπτευόμενης μάθησης (Supervised Learning), θα χρειαζόμασταν ένα ζευγάρι εικόνων εισόδου, μια πρωτότυπη εικόνα και μια αναπαράσταση αυτής της εικόνας στο στυλ που μας ενδιαφέρει. Με αυτές τις εικόνες ένα μοντέλο μηχανικής μάθησης θα μπορούσε να μάθει τον μετασχηματισμό και να τον εφαρμόσει σε διαφορετικές εικόνες.

Αυτή η προσέγγιση όμως δεν είναι εφικτή, καθώς αυτά τα ζευγάρια εικόνων σπάνια υπάρχουν. Για τον λόγο αυτό, τα τελευταία χρόνια έχει αναπτυχθεί η προσέγγιση της μεταφοράς στυλ (Neural Style Transfer - NST), η οποία χρησιμοποιεί βαθιά νευρωνικά δίκτυα τα οποία μπορούν να εξάγουν περιγραφικά χαρακτηριστικά από τις εικόνες περιεχομένου και στυλ ώστε να μπορούν να γίνουν οι απαραίτητοι μετασχηματισμοί για την μεταφορά στυλ, χωρίς να χρειάζονται τα ζευγάρια που θα ήταν απαραίτητα για την προσέγγιση εποπτευόμενης μάθησης.

Για αυτήν την προσέγγιση, συνήθως χρησιμοποιείται ένα μεγάλο και σύνθετο προ-εκπαιδευμένο μοντέλο για την εξαγωγή χαρακτηριστικών και ένα μικρότερο δίκτυο κωδικοποιητή-αποκωδικοποιητή (Encoder-decoder network) για να κάνει την μεταφορά. ([Johnson, Alahi and Fei-Fei, 2016](#))

Αυτό σημαίνει πως για αν επιτευχθεί η μεταφορά στυλ με αυτή την προσέγγιση χρειάζεται μόνο μια εικόνα, η οποία αντιπροσωπεύει το οποίο θέλουμε να μεταφέρουμε σε οποιαδήποτε άλλη εικόνα. Το μειονέκτημα όμως είναι πως με αυτόν τον τρόπο κάθε δίκτυο που εκπαιδεύεται μπορεί να πετύχει μεταφορά μόνο σε ένα συγκεκριμένο στυλ. Αν, για παράδειγμα, έχουμε ένα μοντέλο το οποίο έχει εκπαιδευθεί στα έργα του van Gogh, δεν μπορεί να παράγει εικόνες στο στυλ του Picasso, χωρίς να γίνει επανεκπαίδευση του δικτύου.

Αργότερα αναπτύχθηκαν και μοντέλα τα οποία μπορούν να παράγουν εικόνες σε πολλαπλά στυλ και ακόμα και να συνδυάζει μερικά στυλ για να παράγει ένα καινούργιο. Για να γίνει αυτό, μαζί με την εικόνα περιεχομένου, δίνεται ως είσοδος στο μοντέλο και ένα διάνυσμα που λέει στο δίκτυο πόσο από το κάθε στυλ να εφαρμόσει σε κάθε εικόνα. Με αυτόν τον τρόπο δεν χρειάζεται πλέον ένα μοντέλο για κάθε διαφορετικό στυλ, παρέχοντας έτσι δημιουργική ελευθερία για την ανάμειξη και πειραματισμό με πολλά στυλ. ([Dumoulin, Shlens and Kudlur, 2016](#))

Και οι δυο παραπάνω προσεγγίσεις όμως έχουν τον ίδιο περιορισμό. Μπορούν να παράγουν εικόνες μόνο σε στυλ που έχουν δει κατά την διαδικασία της εκπαίδευσης. Γι' αυτό τον λόγο δημιουργήθηκαν μοντέλα αυθαίρετης μεταφοράς στυλ (Arbitrary Style Transfer), τα οποία στην ουσία μαθαίνουν να εξάγουν και να εφαρμόζουν οποιοδήποτε στυλ σε κάποια εικόνα. ([Huang and Belongie, 2017](#))

Οι τεχνικές μεταφοράς στυλ χρησιμοποιούνται σε διάφορους τομείς, από εμπορική τέχνη μέχρι και εικονική πραγματικότητα και συνεχίζουν να βελτιώνονται με ραγδαίους ρυθμούς.

Στην εργασία αυτή πειραματιζόμαστε με την μεταφορά στυλ μεταξύ των τεχνικών σχεδίασης κινουμένων σχεδίων του ανατολικού κόσμου ("cartoon") και του δυτικού κόσμου ("anime") χρησιμοποιώντας βαθιά συνελκτικά νευρωνικά δίκτυα και την πρώτη τεχνική που απαιτεί την χρήση μιας εικόνας περιεχομένου και μιας εικόνας στυλ.

Κεφάλαιο 4 - Πειραματική Αξιολόγηση

Όπως και έχει αναφερθεί νωρίτερα, η παρούσα διπλωματική ασχολείται με δυο θέματα μηχανικής μάθησης:

- Αναγνώριση και ταξινόμηση στυλ
- Μεταφορά στυλ

Για τον πρώτο στόχο μας έχουν δημιουργηθεί 3 μοντέλα:

- Ένα CNN μοντέλο με 4 επίπεδα συνέλιξης
- Ένα VGG16 μοντέλο
- Ένα VGG19 μοντέλο

Τα μοντέλα αυτά έχουν δημιουργηθεί σε Python, χρησιμοποιώντας διάφορες βιβλιοθήκες ανοιχτού κώδικα, όπως είναι η TensorFlow ή η Keras, οι οποίες χρησιμοποιούνται σε εφαρμογές μηχανικής μάθησης. Όλα τα μοντέλα εκπαιδεύονται στο ίδιο σύνολο δεδομένων και αξιολογούνται με τις ίδιες μετρικές.

Για τον δεύτερο στόχο μας δημιουργήθηκε ένα VGG19 μοντέλο, από το οποίο χρησιμοποιούμε συγκεκριμένα ενδιάμεσα στρώματα για να φτιαχτεί ένα καινούργιο μοντέλο το οποίο θα παίρνει εικόνες και θα παράγει την σύνθεσή τους.

Επιλέχθηκαν 2 εικόνες από το σύνολο δεδομένων που χρησιμοποιήθηκε και για τον προηγούμενο στόχο, πάνω στις οποίες εφαρμόζουμε τις τεχνικές μεταφοράς στυλ για να δούμε τον βαθμό επιτυχίας.

4.1 Αναγνώριση και ταξινόμηση στυλ

4.1.1 Σύνολο δεδομένων

Για την επιλογή του συνόλου δεδομένων υπάρχουν διάφορα κριτήρια τα οποία πρέπει να ληφθούν υπόψιν. Αφού ο στόχος μας είναι η διαφοροποίηση των χαρακτηριστικών και στυλ μεταξύ των ανατολικών και δυτικών τεχνικών σχεδίασης κινουμένων σχεδίων χρειαζόμαστε όσο το δυνατόν περισσότερα καθαρά δείγματα από διάφορες σειρές. Αυτό σημαίνει πως στο σύνολο δεδομένων μας δεν μπορούν να υπάρχουν εικόνες ή ζωγραφιές που έχουν δημιουργηθεί από τρίτους, καθώς έτσι προσθέτουμε και άλλους παράγοντες, όπως είναι η ικανότητα του καθενός στο

να αναπαράγει το στυλ του αρχικού δημιουργού της κάθε σειράς ή την ερμηνεία και τον τρόπο που αντιλαμβάνεται κάποιος τον κόσμο και τους χαρακτήρες.

Για τους λόγους αυτούς αποφασίσαμε πως η καλύτερη επιλογή είναι αντί να χρησιμοποιήσουμε κάποιο από τα διάφορα σύνολα δεδομένων που χρησιμοποιούνται σε διάφορες εφαρμογές, να δημιουργήσουμε ένα δικό μας σύνολο δεδομένων.

Για να επιτευχθεί αυτό διαλέξαμε 40 βίντεο, 20 από κάθε κατηγορία, από διάφορες σειρές, τα οποία αντιπροσωπεύουν διάφορες εποχές, τεχνικές και τεχνολογίες που χρησιμοποιήθηκαν ανά τα χρόνια για την δημιουργία κινουμένων σχεδίων. Τα βίντεο αυτά χωρίστηκαν σε περίπου 250 καρέ (frames) ανά σειρά με αποτέλεσμα να έχουμε ένα σύνολο δεδομένων με σχεδόν 10000 εικόνες (5000 ανά κατηγορία).

Αναφέρεται ότι κατά την διαδικασία της εκπαίδευσης το σύνολο δεδομένων χωρίζεται σε train, test και validation sets με ποσοστό 60%/20%/20%.

Εκτός από αυτό, έχει δημιουργηθεί και ένα δεύτερο μικρό τεστ σύνολο, το οποίο περιλαμβάνει εικόνες όχι μόνο από σειρές που χρησιμοποιήθηκαν για την εκπαίδευση, αλλά και από άλλες σειρές, για να δούμε αν τα μοντέλα μπορούν να κάνουν καλές προβλέψεις σε καινούργια δεδομένα.

4.1.2 Τρόποι Αξιολόγησης

Για το πρώτο μέρος χρησιμοποιήθηκαν οι μετρικές συνολική ακρίβεια (Accuracy), ακρίβεια (Precision), ανάκληση (Recall) και F1-score, στο σύνολο εκπαίδευσης και στο σύνολο τεστ. Αναφέρεται ότι ως θετική κλάση έχει οριστεί η κλάση “cartoon”.

Εκτός από αυτά έχει σημειωθεί η μεταβολή της συνολικής ακρίβειας και της απώλειας του κάθε μοντέλου κατά την διάρκεια της εκπαίδευσης και έχει δημιουργηθεί και ο αντίστοιχος πίνακας σύγχυσης (Confusion Matrix) του κάθε μοντέλου και για το σύνολο εκπαίδευσης αλλά και για το σύνολο τεστ.

4.1.2.1 Συνολική ακρίβεια

Η συνολική ακρίβεια (Accuracy) απαντά στο ερώτημα “Πόσο συχνά έγιναν σωστές προβλέψεις”. Ορίζεται ως:

$$\frac{\text{Αριθμός σωστών προβλέψεων}}{\text{Συνολικός αριθμός προβλεψεων}} \text{ ή αλλιώς } \frac{TP + TN}{TP + FP + TN + FN}$$

4.1.2.2 Ακρίβεια

Η ακρίβεια (Precision) απαντά στο ερώτημα “Από όσα στοιχεία ταξινομήθηκαν στην θετική κλάση, πόσα όντως ανήκουν σε αυτή;”. Ορίζεται ως:

$$\frac{\text{Αριθμός αληθώς θετικών προβλέψεων}}{\text{Αριθμός θετικών προβλέψεων}} \text{ ή αλλιώς } \frac{TP}{TP + FP}$$

4.1.2.3 Ανάκληση

Η ανάκληση (Recall) απαντά στο ερώτημα “Από όσα στοιχεία ανήκουν στην θετική κλάση, πόσα αναγνωρίσαμε σωστά;”. Ορίζεται ως:

$$\frac{\text{Αριθμός αληθώς θετικών προβλέψεων}}{\text{Αριθμός στοιχείων που ανήκουν στην θετική κλάση}} \text{ ή αλλιώς } \frac{TP}{TP + FN}$$

4.1.2.4 F1-score

Το F1 score είναι ο σταθμισμένος μέσος όρος της ακρίβεια και της ανάκλασης. Αυτό σημαίνει πως δίνει ίση βαρύτητα και στις δύο μετρικές. Ορίζεται ως:

$$2 * \frac{\text{Ακρίβεια} * \text{Ανάκληση}}{\text{Ακρίβεια} + \text{Ανάκληση}}$$

4.1.2.5 Πίνακας σύγχυσης

Ο πίνακας σύγχυσης (Confusion Matrix) είναι ένας πίνακας που χρησιμοποιείται για να περιγράψει την απόδοση κάποιου αλγορίθμου ή μοντέλου ταξινόμησης. Κάθε σειρά του πίνακα αντιπροσωπεύει πλήθος των στοιχείων που υπάρχουν στις πραγματικές κλάσεις, ενώ κάθε στήλη αντιπροσωπεύει το πλήθος των στοιχείων που υπάρχουν στις προβλεπόμενες κλάσεις. Σε ένα πρόβλημα δυαδικής αναγνώρισης έχουμε μια θετική (Positive) κλάση και μια αρνητική (Negative) κλάση. Για κάθε μια από τις δυο κλάσεις έχουμε και τους χαρακτηρισμούς της αληθής (True) ή ψευδής (False) ταξινόμησης. Έτσι σχηματίζονται οι ονομασίες των κελιών ανάλογα με την κλάση που αναγνωρίστηκε καθώς και το αν η αναγνώριση αυτή ισχύει. Τα κελιά έχουν τις ονομασίες:

- Αληθώς θετικό (True Positive - TP)
- Ψευδώς θετικό (False Positive - FP)
- Αληθώς αρνητικό (True Negative - TN)
- Ψευδώς αρνητικό (False Negative - FN)

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Σχήμα 8: Παράδειγμα πίνακα σύγχυσης

Στο παράδειγμα αυτό βλέπουμε έναν πίνακα σύγχυσης μαζί με τις ονομασίες των κελιών. (Πηγή εικόνας: <https://www.r-bloggers.com/2020/12/weighting-confusion-matrices-by-outcomes-and-observations-2/>)

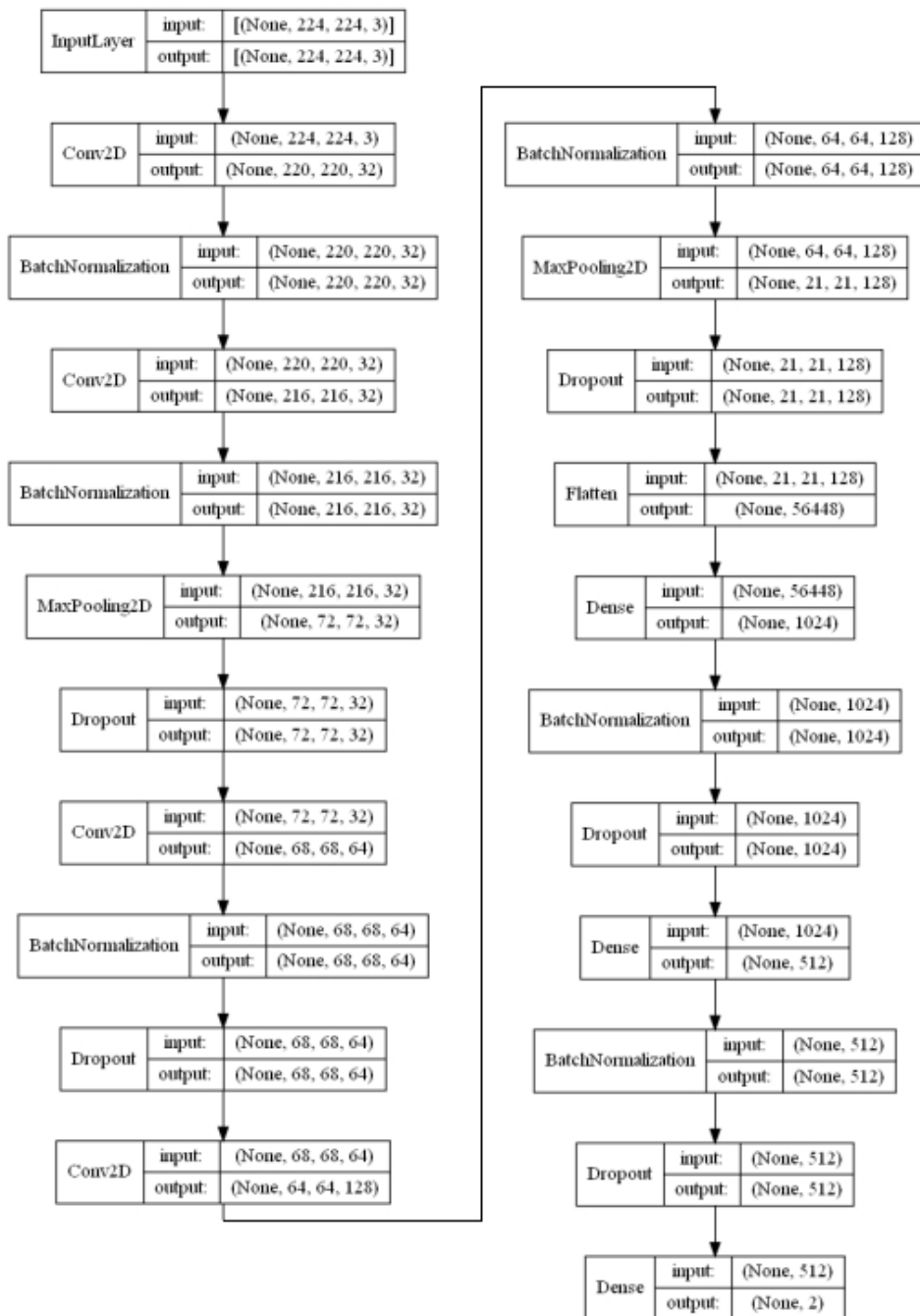
4.1.3 Αρχιτεκτονικές μοντέλων

Τα τρία μοντέλα που δημιουργήθηκαν για την διπλωματική παίρνουν ως είσοδο έγχρωμες εικόνες μεγέθους 224x224, το οποίο αντιστοιχεί σε ένα tensor (224, 224, 3). Τα ενδιάμεσα και τελευταία στρώματα διαφέρουν ανά μοντέλο, με σκοπό να δούμε πως επηρεάζουν το στρώμα εξόδου, το οποίο είναι το ίδιο και για όλα τα μοντέλα, αφού καταλήγουν σε ταξινόμηση ανάμεσα σε δύο διαφορετικές κλάσεις. Η επιλογή των διάφορων παραμέτρων και συναρτήσεων ενεργοποίησης των στρωμάτων έγιναν πειραματικά σε συνδυασμό με διερεύνηση διαφόρων άρθρων στο διαδίκτυο.

4.1.3.1 CNN μοντέλο με 4 επίπεδα συνέλιξης

Το μοντέλο cnn4 δημιουργήθηκε με μια σχετικά απλή αρχιτεκτονική ώστε να υπάρχει ένα μέτρο σύγκρισης με τα πιο σύνθετα VGG μοντέλα. Λόγο της πιο απλής αρχιτεκτονικής περιμένουμε

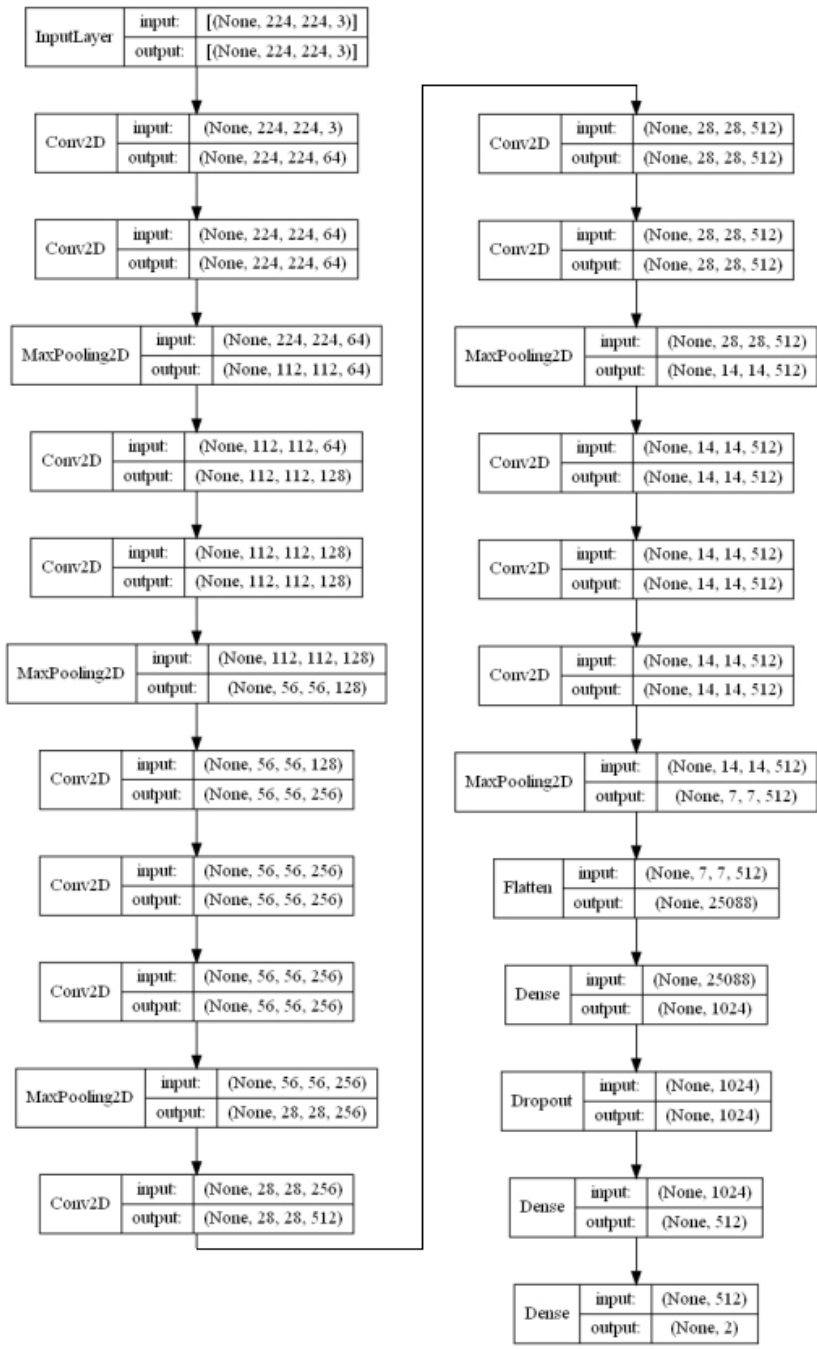
η εκπαίδευση να απαιτεί λιγότερους πόρους, λιγότερο χρόνο αλλά και να παράγει χειρότερα αποτελέσματα (Le, 2018). Η αρχιτεκτονική του παρουσιάζεται παρακάτω:



Σχήμα 9: Αρχιτεκτονική μοντέλου CNN4

4.1.3.2 VGG16

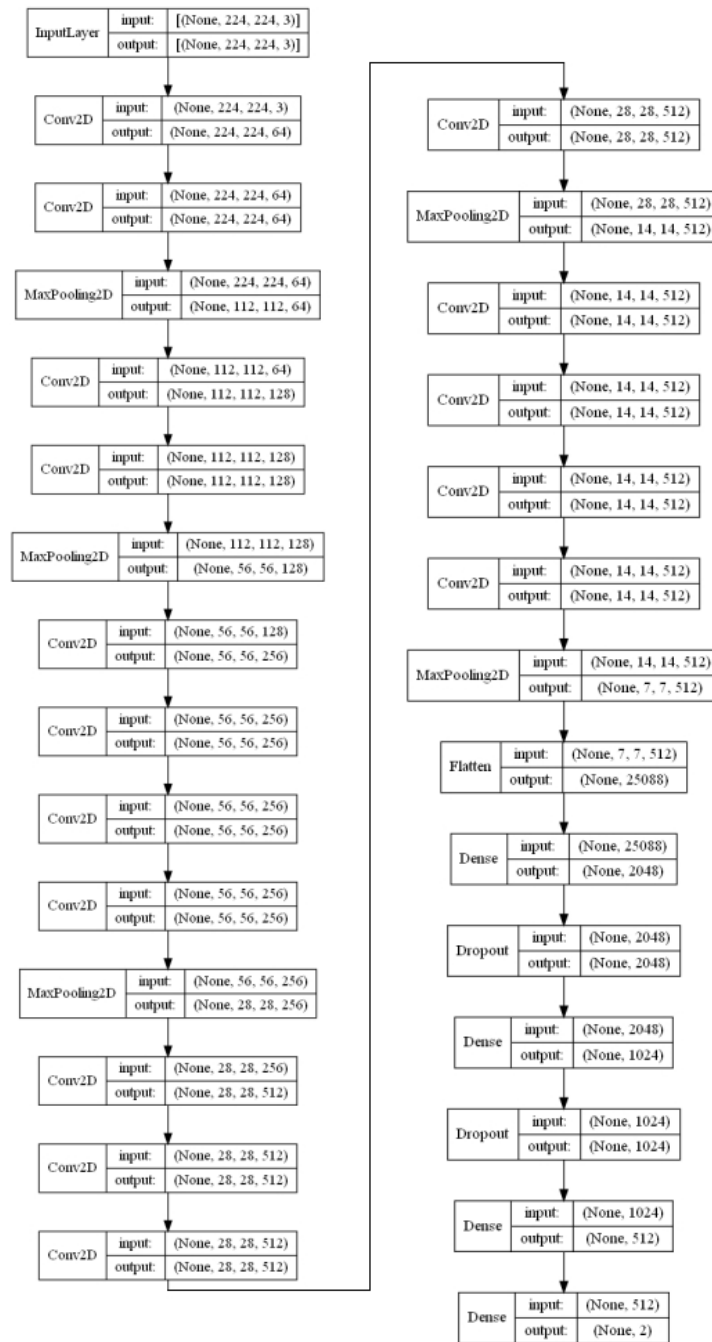
Το μοντέλο VGG16 δημιουργήθηκε με την βοήθεια της βιβλιοθήκης Keras, η οποία παρέχει διάφορες λειτουργίες σχετική με την δημιουργία, εκπαίδευση και αξιολόγηση μοντέλων. Προστέθηκαν στο τέλος μερικά στρώματα ώστε να φτάσουμε στην τελική έξοδο των δύο πιθανών κλάσεων. Η αρχιτεκτονική του παρουσιάζεται παρακάτω:



Σχήμα 10: Αρχιτεκτονική μοντέλου VGG16

4.1.3.3 VGG19

Όπως και το προηγούμενο μοντέλο, το μοντέλο VGG19 δημιουργήθηκε με την βιβλιοθήκη Keras και προστέθηκαν αντίστοιχα στρώματα στο τέλος. Αναφέρεται πως σε αυτό το μοντέλο έχουν αλλάξει και μερικές παράμετροι σε έναν αριθμό ενδιάμεσων στρωμάτων, για να δούμε τον βαθμό που θα επηρεαστούν τα αποτελέσματα. Η αρχιτεκτονική του παρουσιάζεται παρακάτω:



4.1.4 Αποτελέσματα

Σε όλα τα μοντέλα έχουν χρησιμοποιηθεί τεχνικές βελτιστοποίησης, όπως είναι ο βελτιστοποιητής (Adam Optimizer), πρόωρη διακοπή (Early Stopping) και αύξηση δεδομένων (Data Augmentation) και σημείο ελέγχου μοντέλου (Model Checkpoint) με τις ίδιες παραμέτρους. Τα παραπάνω έχουν οδηγήσει σε διαφορετικό αριθμό εποχών εκπαίδευσης, με βάση το πότε έκρινε η πρόωρη διακοπή κατάλληλο να σταματήσει η εκπαίδευση και το μοντέλο που κρατήθηκε σε κάθε περίπτωση είναι αυτό το οποίο είχε τις καλύτερες επιδόσεις με βάση την παράμετρο ελέγχου του σημείο ελέγχου μοντέλου.

4.1.4.1 CNN μοντέλο με 4 επίπεδα συνέλιξης

Όπως ήταν αναμενόμενο, η σχετικά πιο απλή αρχιτεκτονική του μοντέλου cnn4 οδήγησε σε μικρότερο χρόνο εκπαίδευσης ανά εποχή, χρησιμοποιώντας λιγότερους πόρους σε σχέση με τα άλλα μοντέλα, αλλά τα αποτελέσματα δεν είναι ιδιαίτερα καλά, ειδικά για ταξινόμηση δύο κλάσεων. Βλέπουμε μια ξεκάθαρη τάση προς την θετική κλάση, πράγμα που οφείλεται σε υπερεκπαίδευση κυρίως λόγω του συνόλου δεδομένων.

4.1.4.1.1 Σύνολο εκπαίδευσης

Τα αποτελέσματα του κώδικα όπως εμφανίζονται στην κονσόλα:

```
Classification report
      precision    recall  f1-score   support

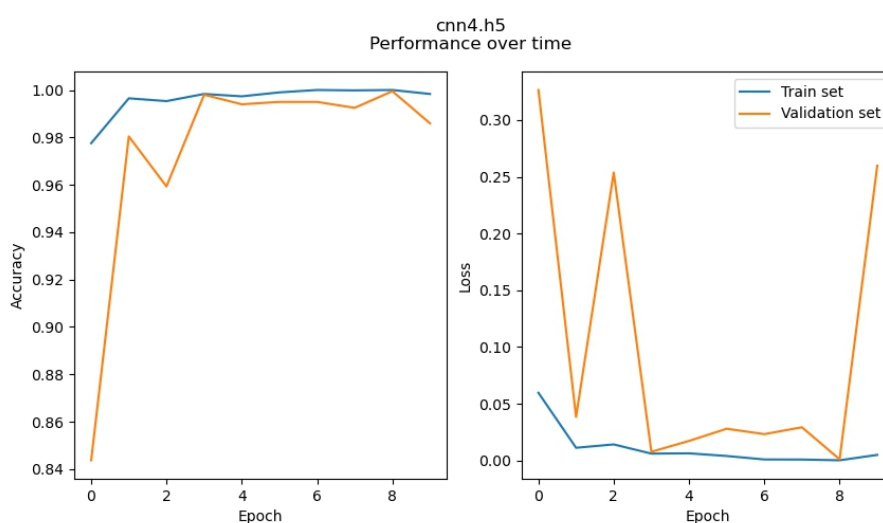
   anime      1.00      0.04      0.07     1023
   cartoon    0.50      1.00      0.66      967

 accuracy          0.51     1990
 macro avg      0.75      0.52      0.37     1990
 weighted avg   0.75      0.51      0.36     1990

cnn4.h5 metrics:
Accuracy: 50.55% Precision: 49.56% Recall: 100.00% F1: 66.28%
```

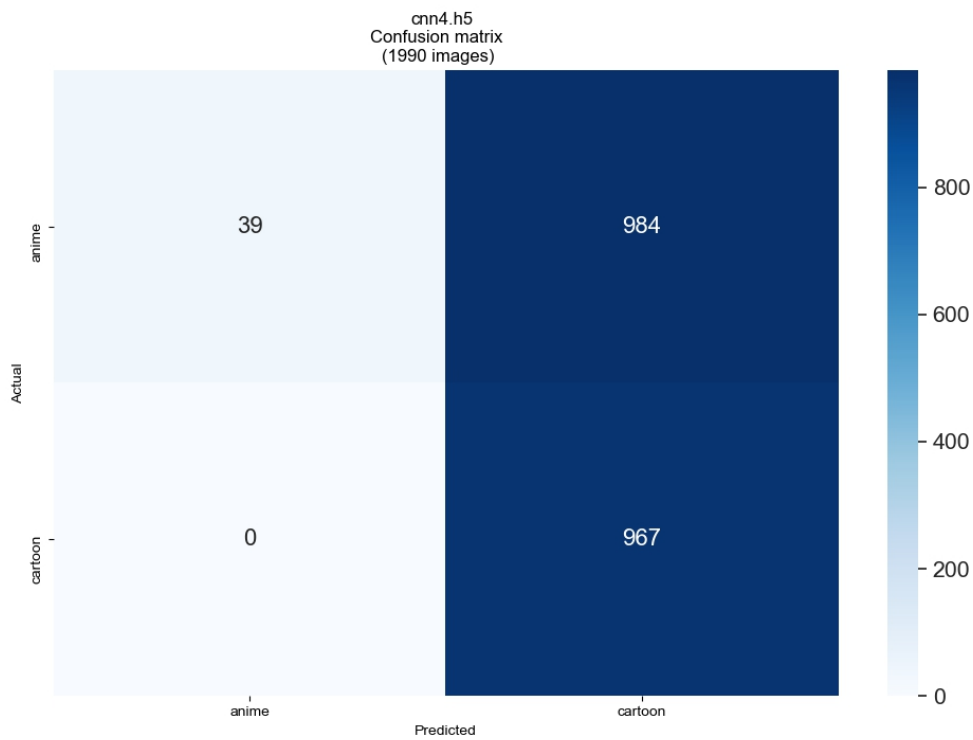
Σχήμα 11: Αποτελέσματα κώδικα ρυθμον του μοντέλου cnn4

Η απόδοση του μοντέλου κατά την διάρκεια εκπαίδευσης:



Σχήμα 12: Διαγράμματα συνολικής ακρίβειας και απώλειας του μοντέλου cnn4

Ο πίνακας σύγχυσης:



Σχήμα 13: Πίνακας σύγχυσης (σύνολο εκπαίδευσης) του μοντέλου cnn4

4.1.4.1.2 Σύνολο τεστ

Τα αποτελέσματα του κώδικα όπως εμφανίζονται στην κονσόλα:

```
Classification report
      precision    recall  f1-score   support

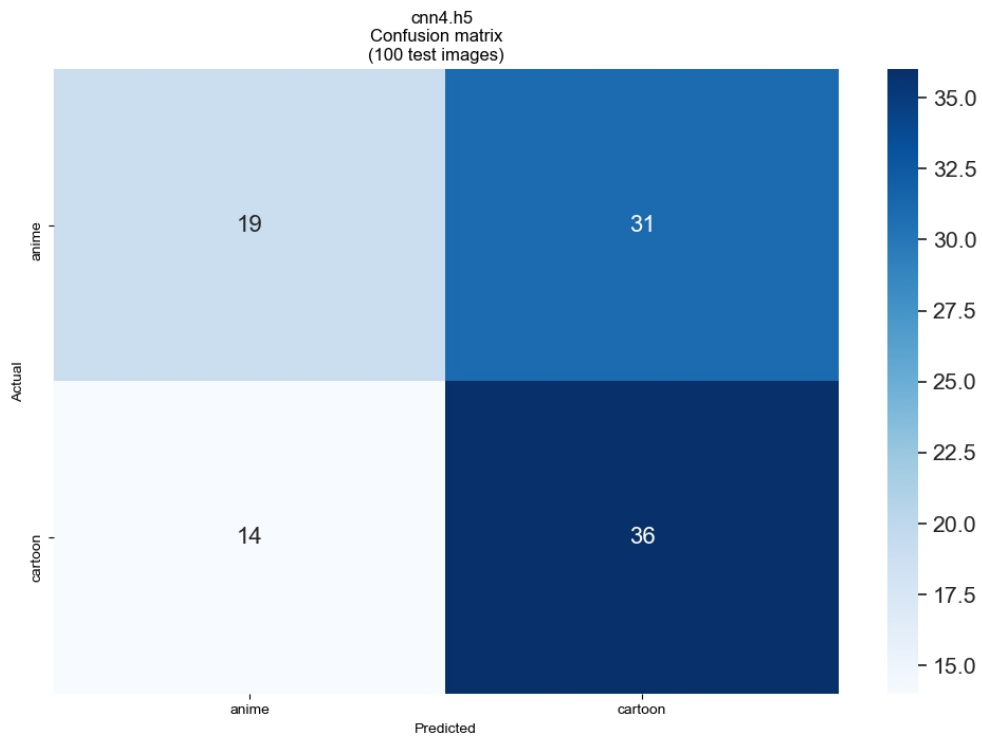
   anime      0.58      0.38      0.46         50
   cartoon    0.54      0.72      0.62         50

 accuracy      0.55         100
 macro avg     0.56      0.55      0.54         100
weighted avg     0.56      0.55      0.54         100

cnn4.h5 metrics (test images)
Accuracy: 55.00% Precision: 53.73% Recall: 72.00% F1: 61.54%
```

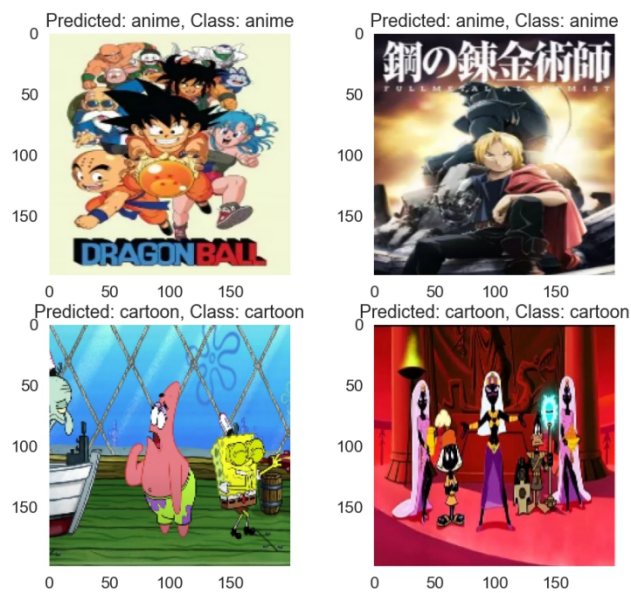
Σχήμα 14: Αποτελέσματα classifier του μοντέλου cnn4

Ο πίνακας σύγχυσης:

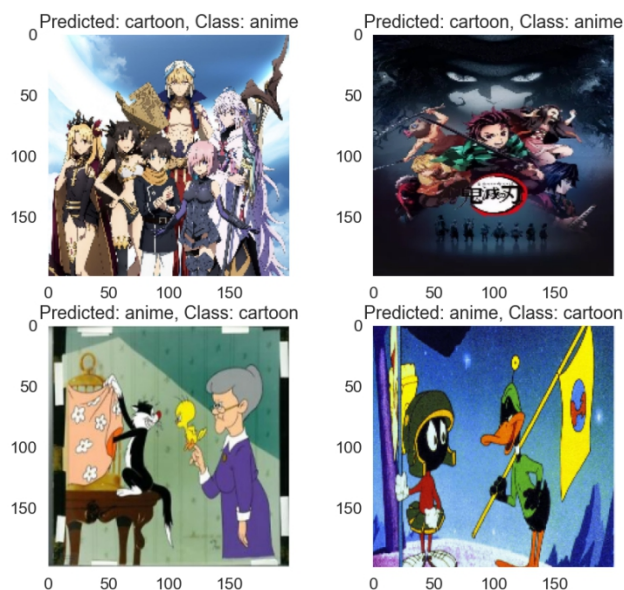


Σχήμα 15: Πίνακας σύγχυσης (σύνολο τεστ) του μοντέλου cnn4

Παραδείγματα σωστής και λανθασμένης αναγνώρισης συλ:



Σχήμα 16: Παραδείγματα σωστής αναγνώρισης του μοντέλου cnn4



Σχήμα 17: Παραδείγματα λανθασμένης αναγνώρισης του μοντέλου cnn4

4.1.4.2 VGG16

Η πιο σύνθετη αρχιτεκτονική του μοντέλου VGG16 απαιτεί αρκετά μεγαλύτερο χρόνο εκπαίδευσης ανά εποχή καθώς και υπολογιστικούς πόρους, παράγει όμως πολύ καλύτερα αποτελέσματα σε σχέση με το πιο απλό cnn4 μοντέλο. Παρατηρούμε και εδώ παρόμοια τάση προς την θετική κλάση, όχι όμως τόσο ακραία όσο νωρίτερα.

4.1.4.2.1 Σύνολο εκπαίδευσης

Τα αποτελέσματα του κώδικα όπως εμφανίζονται στην κονσόλα:

```

Classification report
      precision    recall  f1-score   support

   anime         1.00      0.49      0.66      1021
   cartoon        0.65      1.00      0.79       969

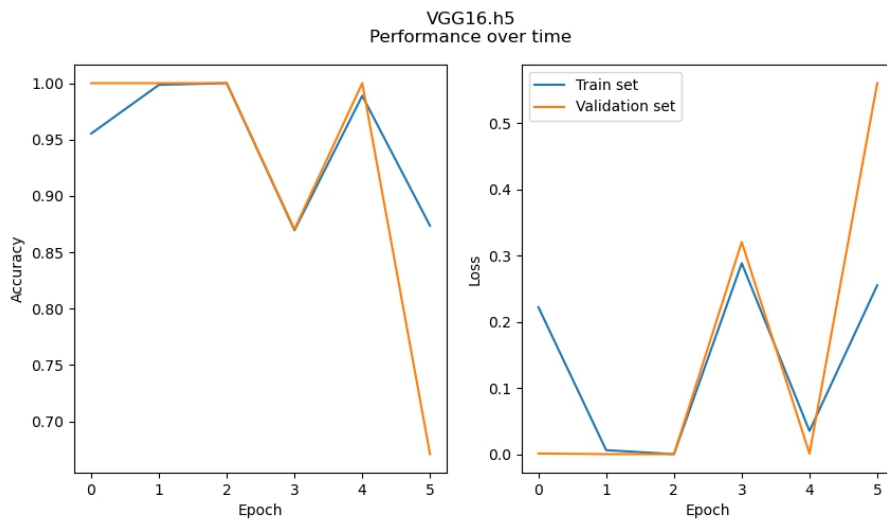
 accuracy                   0.74      1990
 macro avg          0.82      0.74      0.72      1990
 weighted avg       0.83      0.74      0.72      1990

VGG16.h5 metrics:
Accuracy: 73.72% Precision: 64.95% Recall: 100.00% F1: 78.75%

```

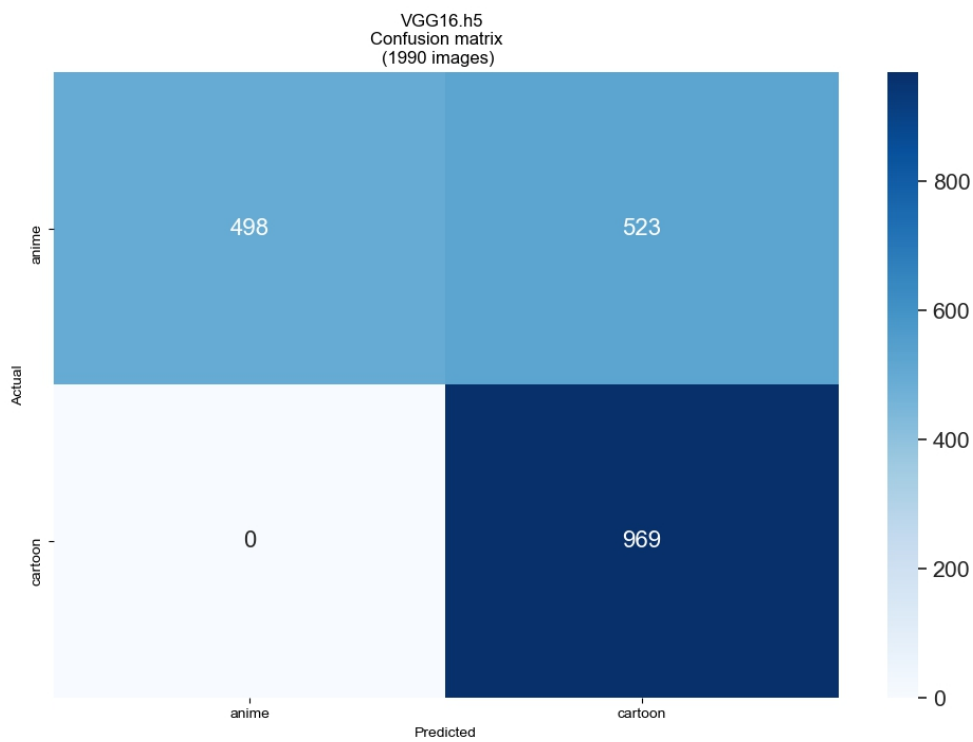
Σχήμα 18: Αποτελέσματα κώδικά ρυθον του μοντέλου VGG16

Η απόδοση του μοντέλου κατά την διάρκεια εκπαίδευσης:



Σχήμα 19: Διαγράμματα συνολικής ακρίβειας και απώλειας του μοντέλου VGG16

Ο πίνακας σύγχυσης:



Σχήμα 20: Πίνακας σύγχυσης (σύνολο εκπαίδευσης) του μοντέλου VGG16

4.1.4.2.2 Σύνολο τεστ

Τα αποτελέσματα του κώδικα όπως εμφανίζονται στην κονσόλα:

```

Classification report
              precision    recall  f1-score   support

   anime      0.50      0.16      0.24      50
   cartoon    0.50      0.84      0.63      50

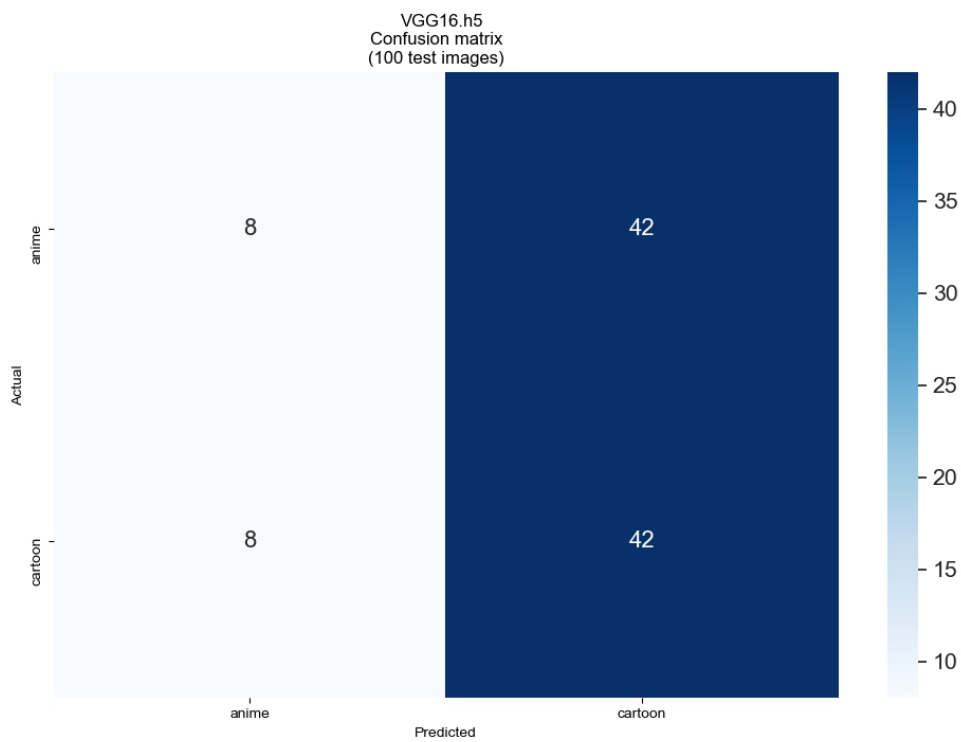
 accuracy      0.50      0.50      0.50      100
 macro avg     0.50      0.50      0.43      100
 weighted avg  0.50      0.50      0.43      100

VGG16.h5 metrics (test images)
Accuracy: 50.00% Precision: 50.00% Recall: 84.00% F1: 62.69%

```

Σχήμα 21: Αποτελέσματα classifier του μοντέλου VGG16

Ο πίνακας σύγχυσης:

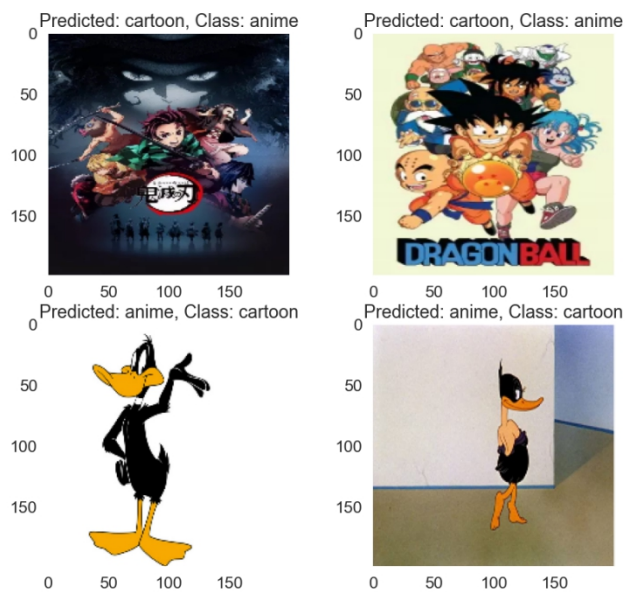


Σχήμα 22: Πίνακας σύγχυσης (σύνολο τεστ) του μοντέλου VGG16

Παραδείγματα σωστής και λανθασμένης αναγνώρισης συλ:



Σχήμα 23: Παραδείγματα σωστής αναγνώρισης του μοντέλου VGG16



Σχήμα 24: Παραδείγματα λανθασμένης αναγνώρισης του μοντέλου VGG16

4.1.4.3 VGG19

Το μοντέλο VGG19, το οποίο έχει παρόμοια, αλλά λίγο πιο σύνθετη αρχιτεκτονική σε σχέση με αυτή του μοντέλου VGG16, έχει παρόμοιες απαιτήσεις χρόνου και υπολογιστικών πόρων με αυτές του VGG16, και σε φυσιολογικές συνθήκες θα περιμέναμε να εμφανίσει σχετικά καλύτερα αποτελέσματα. Παρατηρούμε όμως ότι παρουσιάζει ελάχιστα χειρότερα αποτελέσματα σε σχέση με το μοντέλο VGG16, πράγμα που πιθανά οφείλεται στις διαφορές που αναφέρθηκαν νωρίτερα.

4.1.4.2.1 Σύνολο εκπαίδευσης

Τα αποτελέσματα του κώδικα όπως εμφανίζονται στην κονσόλα:

```
Classification report
              precision    recall  f1-score   support

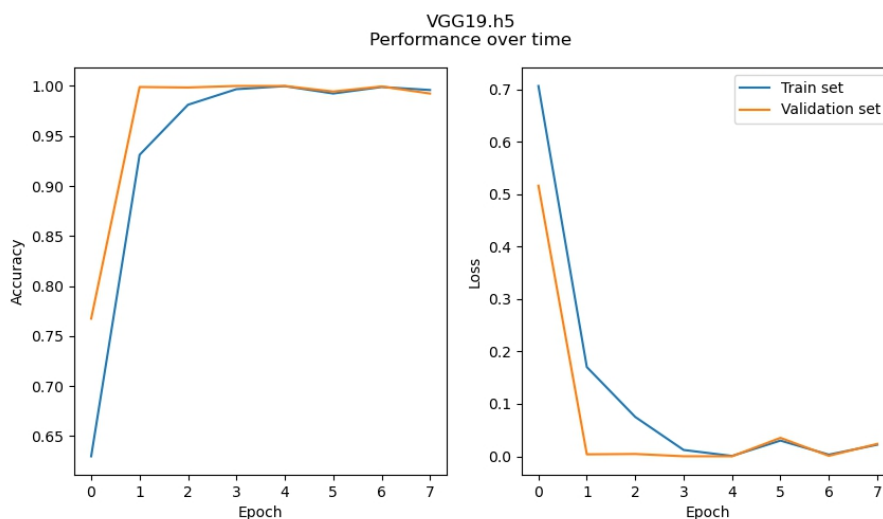
   anime      1.00      0.40      0.58      1019
   cartoon    0.62      1.00      0.76       971

 accuracy                0.69      1990
 macro avg      0.81      0.70      0.67      1990
 weighted avg   0.81      0.69      0.67      1990

VGG19.h5 metrics:
Accuracy: 69.50% Precision: 61.53% Recall: 100.00% F1: 76.19%
```

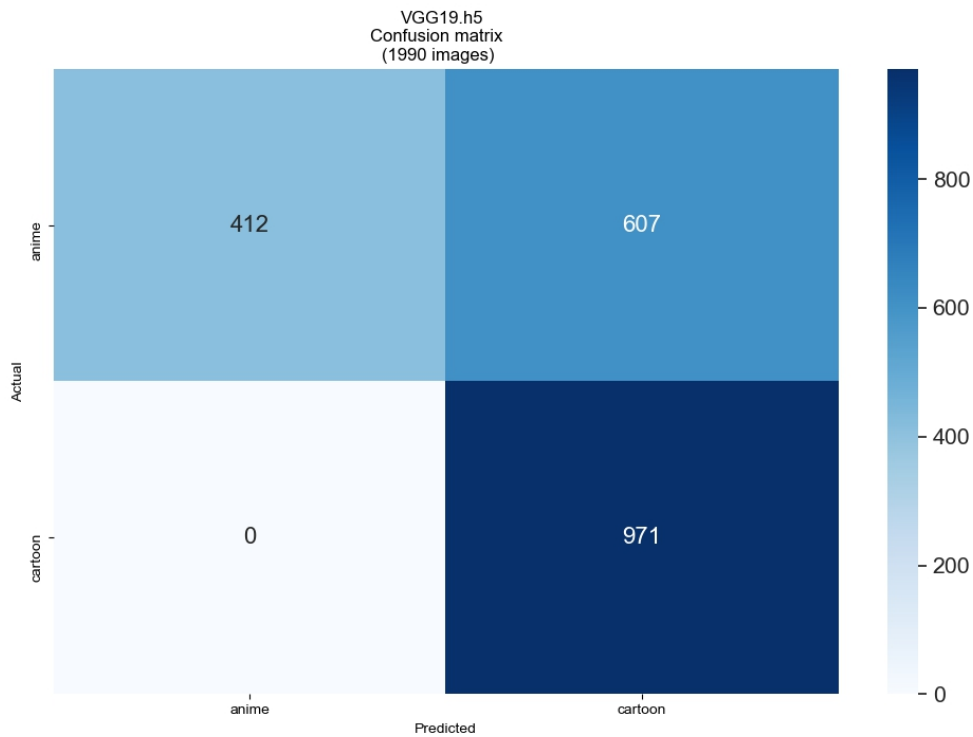
Σχήμα 25: Αποτελέσματα κώδικα rython του μοντέλου VGG19

Η απόδοση του μοντέλου κατά την διάρκεια εκπαίδευσης:



Σχήμα 26: Διαγράμματα συνολικής ακρίβειας και απώλειας του μοντέλου VGG19

Ο πίνακας σύγκρισης:



Σχήμα 27: Πίνακας σύγχυσης (σύνολο εκπαίδευσης) του μοντέλου VGG19

4.1.4.2.2 Σύνολο τεστ

Τα αποτελέσματα του κώδικα όπως εμφανίζονται στην κονσόλα:

```
Classification report
      precision    recall  f1-score   support

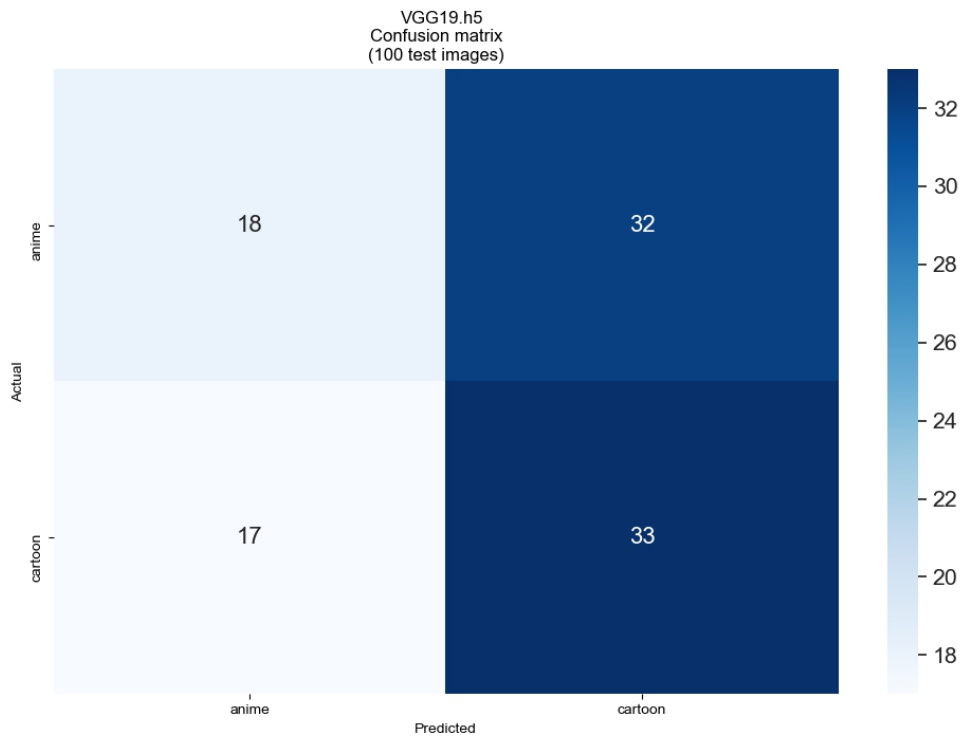
   anime      0.51      0.36      0.42      50
  cartoon      0.51      0.66      0.57      50

 accuracy              0.51      100
 macro avg      0.51      0.51      0.50      100
weighted avg      0.51      0.51      0.50      100

VGG19.h5 metrics (test images)
Accuracy: 51.00% Precision: 50.77% Recall: 66.00% F1: 57.39%
```

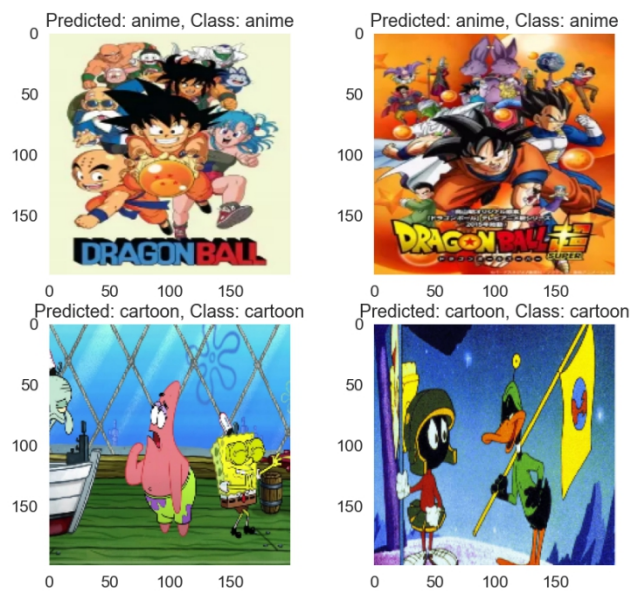
Σχήμα 28: Αποτελέσματα classifier του μοντέλου VGG19

Ο πίνακας σύγχυσης:

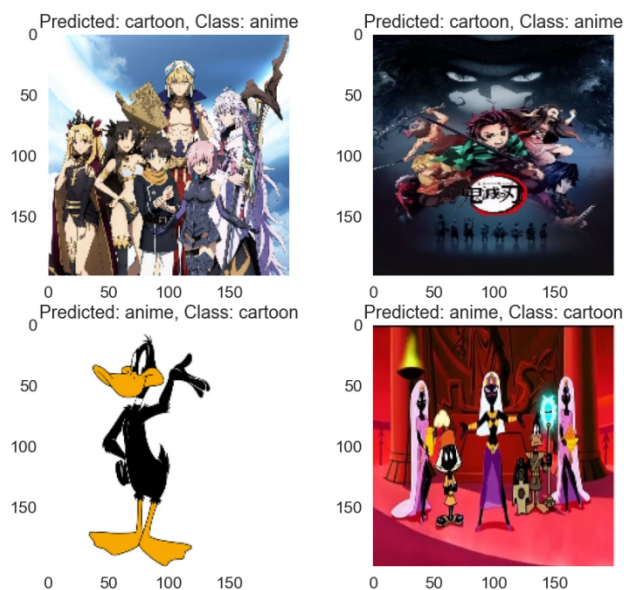


Σχήμα 29: Πίνακας σύγκρισης (σύνολο τεστ) του μοντέλου VGG19

Παραδείγματα σωστής και λανθασμένης αναγνώρισης συλ:

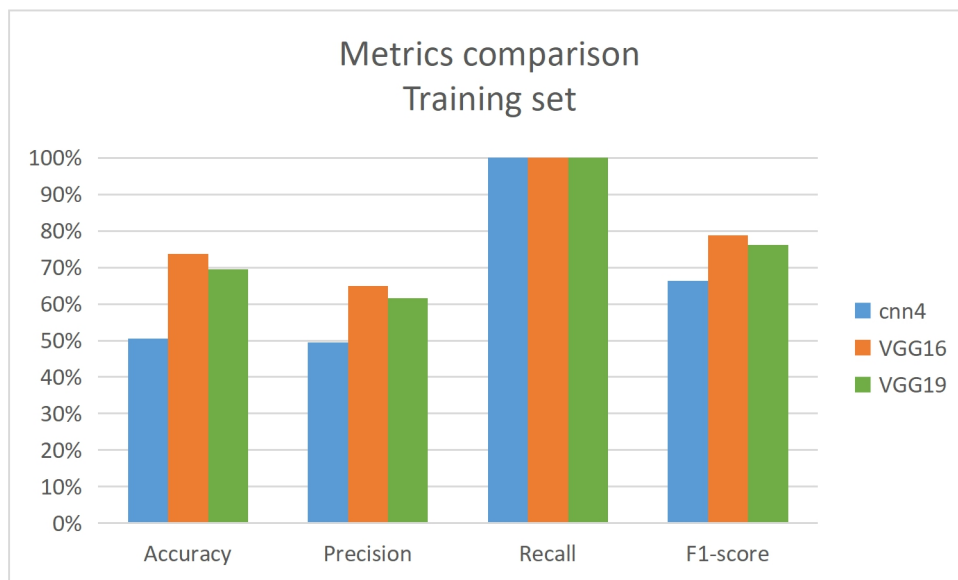


Σχήμα 30: Παραδείγματα σωστής αναγνώρισης του μοντέλου VGG19



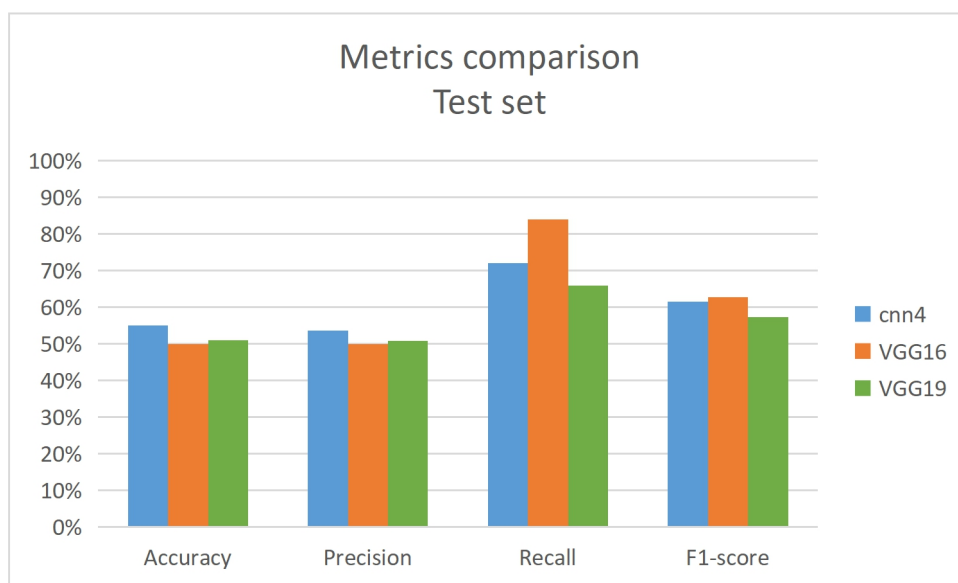
Σχήμα 31: Παραδείγματα λανθασμένης αναγνώρισης του μοντέλου VGG19

4.1.5 Σύγκριση αποτελεσμάτων



Σχήμα 32: Διάγραμμα σύγκρισης αποτελεσμάτων στο σύνολο εκπαίδευσης

Στο σύνολο εκπαίδευσης παρατηρούμε άριστη ανάκληση (100%) σε όλα τα μοντέλα, όμως λόγω της σημαντικής διαφοράς των VGG μοντέλων, στην ακρίβεια, σε σχέση με το cnn4 μοντέλο, βλέπουμε πως το τελικό F1-score διαφέρει αρκετά ανάμεσα στο πιο απλό CNN μοντέλο και τα πιο σύνθετα VGG μοντέλα.



Σχήμα 33: Διάγραμμα σύγκρισης αποτελεσμάτων στο σύνολο τεστ

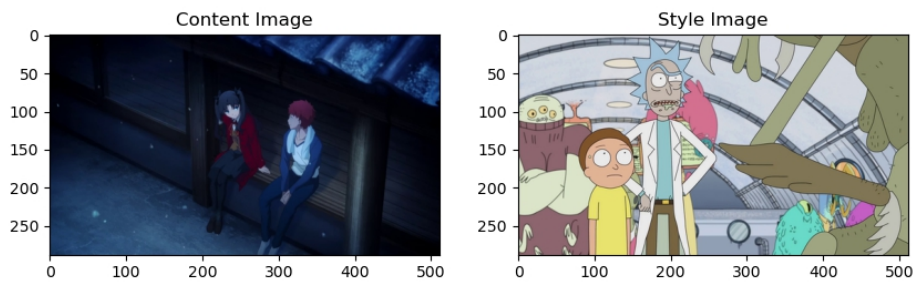
Για το συγκεκριμένο σύνολο τεστ που χρησιμοποιήθηκε σε αυτή την εργασία βλέπουμε ότι και τα τρία μοντέλα πετυχαίνουν παρόμοια αποτελέσματα, με βάση το F1-score. Ενώ στις μετρικές τις ακρίβειας όλα τα μοντέλα πετυχαίνουν παρόμοια ποσοστά, παρατηρούμε ότι σε αυτό το σύνολο η διαφορά εμφανίζεται στην μετρική της ανάκλησης, η οποία δίνει ένα ελάχιστο προβάδισμα στο F1-score του μοντέλου VGG16.

Αξίζει επίσης να σημειωθεί πως το μοντέλο VGG16 φαίνεται να παράγει καλύτερα αποτελέσματα και στα δυο σύνολα, ενώ το μοντέλο VGG19 παράγει καλύτερα αποτελέσματα στο σύνολο εκπαίδευσης, σε σχέση με το μοντέλο cnn4, το οποίο με την σειρά του φαίνεται να έχει καλύτερη επιτυχία με το σύνολο τεστ του πειράματος.

4.2 Μεταφορά στυλ

4.2.1 Σύνολο δεδομένων

Για την μεταφορά στυλ χρησιμοποιώντας ένα VGG19 μοντέλο χρειαζόμαστε μόνο δύο εικόνες, μια από κάθε κατηγορία, τις οποίες επιλέξαμε από το ήδη υπάρχον σύνολο δεδομένων και παρουσιάζονται παρακάτω:



Σχήμα 34: Εικόνες που επιλέχθηκαν για μεταφορά στυλ (Πείραμα 1)

Σε αυτό το πείραμα θα προσπαθήσουμε να παραχωρήσουμε στην αριστερή εικόνα, η οποία έχει τον ρόλο της εικόνας περιεχομένου (Content Image), το στυλ της δεξιάς εικόνας, η οποία έχει τον ρόλο της εικόνας στυλ (Style Image).

Σημειώνεται ότι η αριστερή εικόνα ανήκει στην κατηγορία “anime”, ενώ η αριστερή στην κατηγορία “cartoon”. Δεν είναι απαραίτητο οι εικόνες να ανήκουν σε διαφορετικές κατηγορίες για να γίνει η μεταφορά στυλ, απλά επιλέχθηκαν έτσι καθώς η εργασία αυτή συγκρίνει τις διαφορές μεταξύ των δυο στυλ.

4.2.2 Τρόποι αξιολόγησης

Για την αξιολόγηση των πειραματικών αποτελεσμάτων της μεταφοράς στυλ χρησιμοποιήθηκαν οι μετρικές:

- Συνολική απώλεια (Total Loss)
- Απώλεια περιεχομένου (Content Loss)
- Απώλεια στυλ (Style Loss)

4.2.2.1 Συνολική απώλεια

Η συνολική απώλεια (Total Loss) είναι το άθροισμα της απώλειας περιεχομένου (Content Loss) με την απώλεια στυλ (Style Loss), οι οποίες περιγράφονται παρακάτω.

4.2.2.2 Απώλεια περιεχομένου

Η απώλεια περιεχομένου (Content Loss) είναι μια συνάρτηση που περιγράφει την απόσταση μεταξύ μία εικόνας εισόδου x και μίας εικόνας περιεχομένου p . Δίνεται από τον τύπο:

$$L_{\text{content}}^l(p, x) = \sum_{i,j} (F_{ij}^l(x) - P_{ij}^l(p))^2$$

Όπου τα $F_{ij}^l(x)$ και $P_{ij}^l(p)$ είναι η αντίστοιχη αναπαράσταση χαρακτηριστικών του δικτύου με είσοδο τις εικόνες p και x στο στρώμα l .

Με άλλα λόγια, περνάμε τις εικόνες από το δίκτυο, το οποίο θα επιστρέψει τις ενδιαμέσες αναπαραστάσεις αυτών των εικόνων και υπολογίζουμε την ευκλείδεια απόσταση αυτών.

4.2.2.2 Απώλεια στυλ

Η απώλεια στυλ (Style Loss) υπολογίζεται με παρόμοιο τρόπο, χρησιμοποιώντας μια εικόνα εισόδου και μια εικόνα στυλ. Αντί όμως να συγκρίνουμε τις ενδιαμέσες αναπαραστάσεις αυτών των εικόνων, που παράγονται από τα ενδιαμέσα στρώματα, συγκρίνουμε τους πίνακες Gram (Gram matrices) των εξόδων.

Ορίζουμε την απώλεια στυλ μίας εικόνας εισόδου x και μίας εικόνας στυλ a , ως την απόσταση μεταξύ της αναπαράστασης στυλ (οι πίνακες Gram) αυτών των εικόνων.

Η αναπαράσταση στυλ μιας εικόνας περιγράφεται ως η συσχέτιση μεταξύ των διαφορετικών αποκρίσεων φίλτρων που δίνονται από τον πίνακα Gram G^l , όπου G_{ij}^l είναι το εσωτερικό γινόμενο μεταξύ των χαρτών χαρακτηριστικών i και j στο επίπεδο l . Το G_{ij}^l , το οποίο δημιουργήθηκε από τον χάρτη χαρακτηριστικών μια δεδομένης εικόνας, αντιπροσωπεύει τη συσχέτιση μεταξύ των χαρτών χαρακτηριστικών i και j .

Για να δημιουργήσουμε ένα στυλ για την εικόνα εισόδου, χρησιμοποιούμε την τεχνική της κατάβασης κλίσης (Gradient Descent) από την εικόνα περιεχομένου για να την μετατρέψουμε σε μια εικόνα που ταιριάζει με την αναπαράσταση στυλ της αρχικής εικόνας. Για να το πετύχουμε αυτό πρέπει να ελαχιστοποιήσουμε την μέση τετραγωνική απόσταση μεταξύ του χάρτη χαρακτηριστικών της εικόνας στυλ και της εικόνας εισόδου. Ο βαθμός συνεισφοράς του κάθε επιπέδου περιγράφεται από την σχέση:

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2$$

Όπου τα G_{ij}^l και A_{ij}^l είναι οι αντίστοιχες αναπαραστάσεις στυλ στο στρώμα l της εικόνας εισόδου x και εικόνας στυλ a . Το N_l περιγράφει τον αριθμό των χαρτών χαρακτηριστικών, όπου ο καθένας έχει μέγεθος M_l . Έτσι η απώλεια στυλ περιγράφεται από την σχέση:

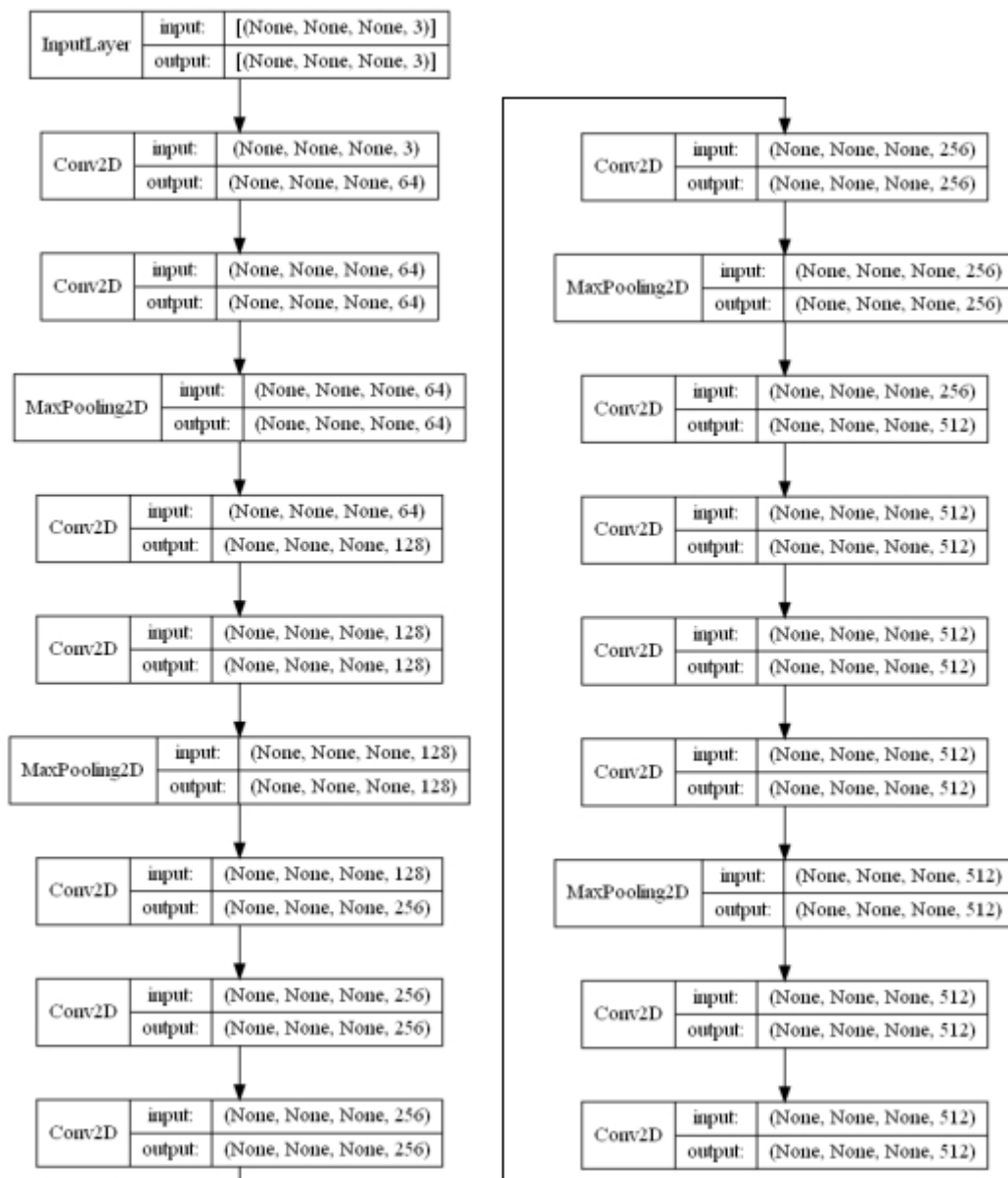
$$L_{\text{style}}(a, x) = \sum_{l \in L} w_l E_l$$

Όπου σταθμίζουμε την συμβολή της απώλειας κάθε στρώματος κατά έναν παράγοντα w_i . Στην συγκεκριμένη περίπτωση κάθε στρώμα έχει ισάξια συμβολή.

4.2.3 Αρχιτεκτονική μοντέλου

Για την μεταφορά στυλ χρησιμοποιούμε ένα VGG19 μοντέλο, από το οποίο μας ενδιαφέρει να κρατήσουμε κάποια από τα ενδιάμεσα στρώματα, στα οποία δημιουργείται και επεξεργάζεται πληροφορία σχετική με την εικόνα περιεχομένου και το στυλ το οποίο προσπαθούμε να της αποδώσουμε. Τα στρώματα αυτά θα χρησιμοποιηθούν ως στρώματα εξόδου σε ένα μοντέλο το οποίο θα παίρνει ως είσοδο εικόνες και ως εξόδους τους χάρτες χαρακτηριστικών και την πληροφορία σχετική με το στυλ που μας ενδιαφέρει. (Surma, 2019; Yuan, 2018; Chaurasia, 2020)

Η αρχιτεκτονική του μοντέλου αυτού παρουσιάζεται παρακάτω:



Σχήμα 35: Αρχιτεκτονική μοντέλου μεταφοράς στυλ

4.2.4 Αποτελέσματα

Τα αποτελέσματα που παρουσιάζονται παρακάτω προκύπτουν μετά από 1000 επαναλήψεις.

Η μεταμόρφωση της αρχικής εικόνας κατά την περίοδο εφαρμογής των τεχνικών μεταφοράς στυλ (κάθε εικόνα παράγεται 100 επαναλήψεις μετά την προηγούμενη):



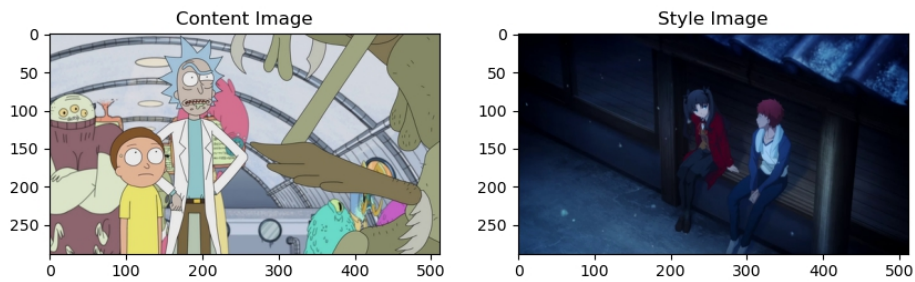
Σχήμα 36: Η σταδιακή μεταμόρφωση της εικόνας περιεχομένου (Πείραμα 1)

Η τελική εικόνα:



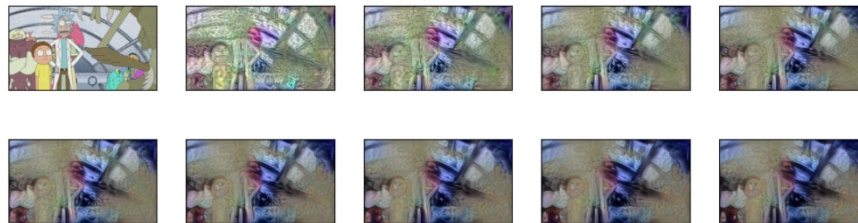
Σχήμα 37: Αποτέλεσμα μεταφοράς στυλ (Πείραμα 1)

Τα αποτελέσματα θα διαφέρουν αν αντιστρέψουμε τον ρόλο των δυο εικόνων. Σε αυτό το πείραμα η εικόνα περιεχομένου ανήκει στην κατηγορία “cartoon” ενώ η εικόνα στυλ την κατηγορία “anime” όπως φαίνεται παρακάτω:



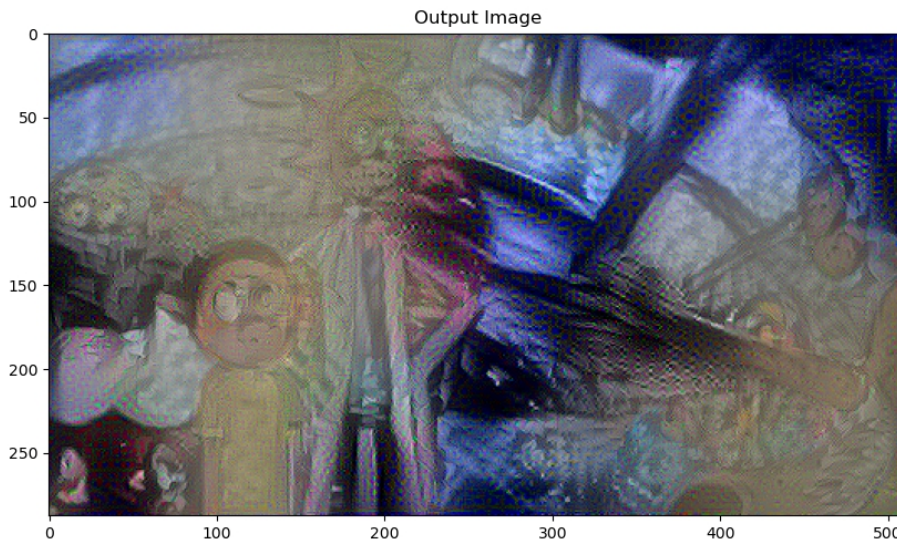
Σχήμα 38: Εικόνες που επιλέχθηκαν για μεταφορά στυλ (Πείραμα 2)

Η μεταμόρφωση της αρχικής εικόνας κατά την περίοδο εφαρμογής των τεχνικών μεταφοράς στυλ (κάθε εικόνα παράγεται 100 επαναλήψεις μετά την προηγούμενη):



Σχήμα 39: Η σταδιακή μεταμόρφωση της εικόνας περιεχομένου (Πείραμα 2)

Η τελική εικόνα:



Σχήμα 40: Αποτέλεσμα μεταφοράς στυλ (Πείραμα 2)

Στην συνέχεια παρουσιάζονται τα αποτελέσματα που προκύπτουν μετά από 10000 επαναλήψεις.

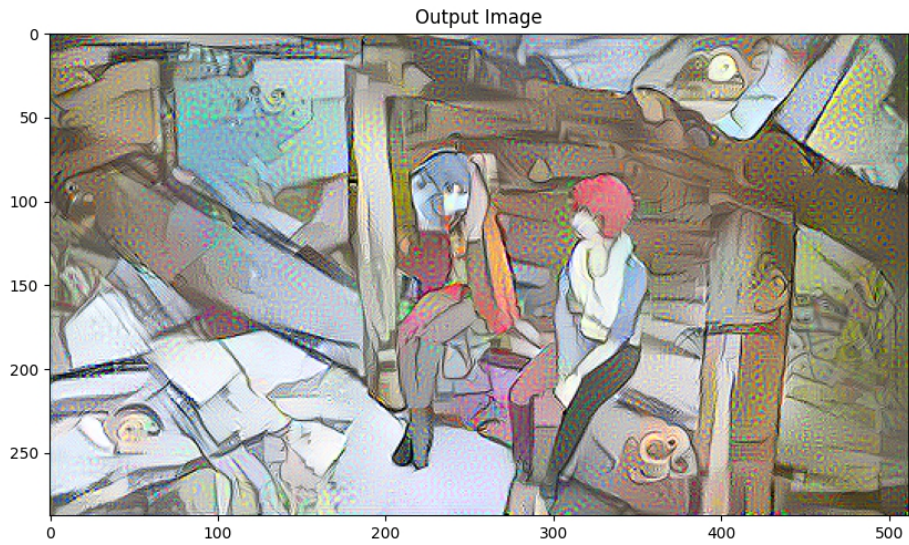
Για την μεταφορά “cartoon” στυλ σε “anime” εικόνα περιεχομένου έχουμε:

Η μεταμόρφωση της αρχικής εικόνας κατά την περίοδο εφαρμογής των τεχνικών μεταφοράς στυλ (κάθε εικόνα παράγεται 1000 επαναλήψεις μετά την προηγούμενη):



Σχήμα 41: Η σταδιακή μεταμόρφωση της εικόνας περιεχομένου (Πείραμα 3)

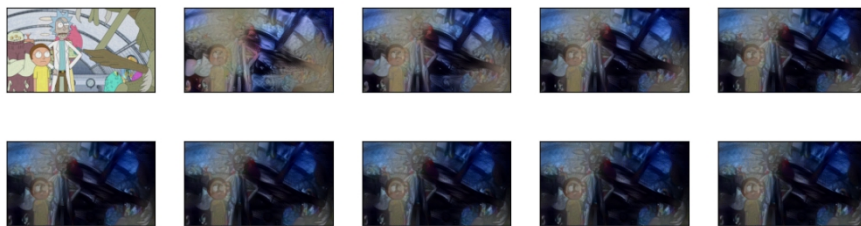
Η τελική εικόνα:



Σχήμα 42: Αποτέλεσμα μεταφοράς στυλ (Πείραμα 3)

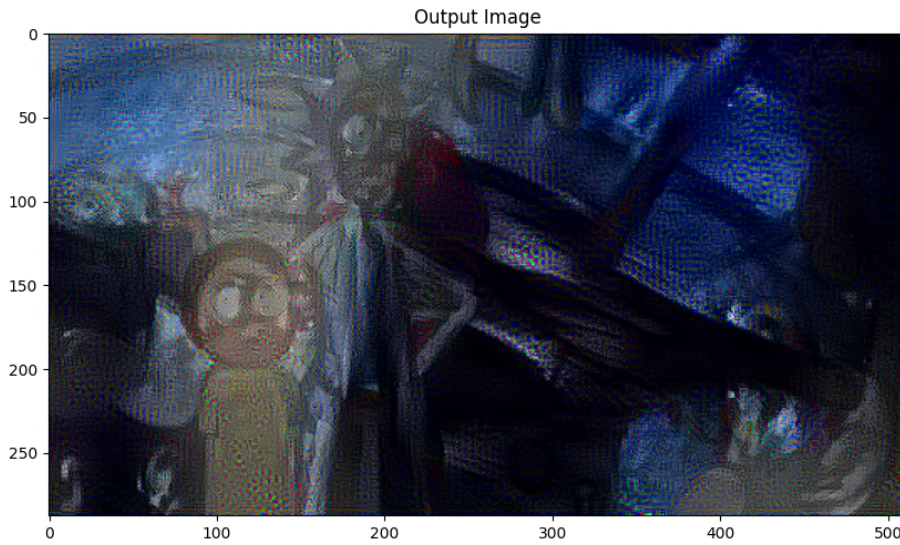
Για την μεταφορά “anime” στυλ σε “cartoon” εικόνα περιεχομένου έχουμε:

Η μεταμόρφωση της αρχικής εικόνας κατά την περίοδο εφαρμογής των τεχνικών μεταφοράς στυλ (κάθε εικόνα παράγεται 1000 επαναλήψεις μετά την προηγούμενη):



Σχήμα 43: Η σταδιακή μεταμόρφωση της εικόνας περιεχομένου (Πείραμα 4)

Η τελική εικόνα:



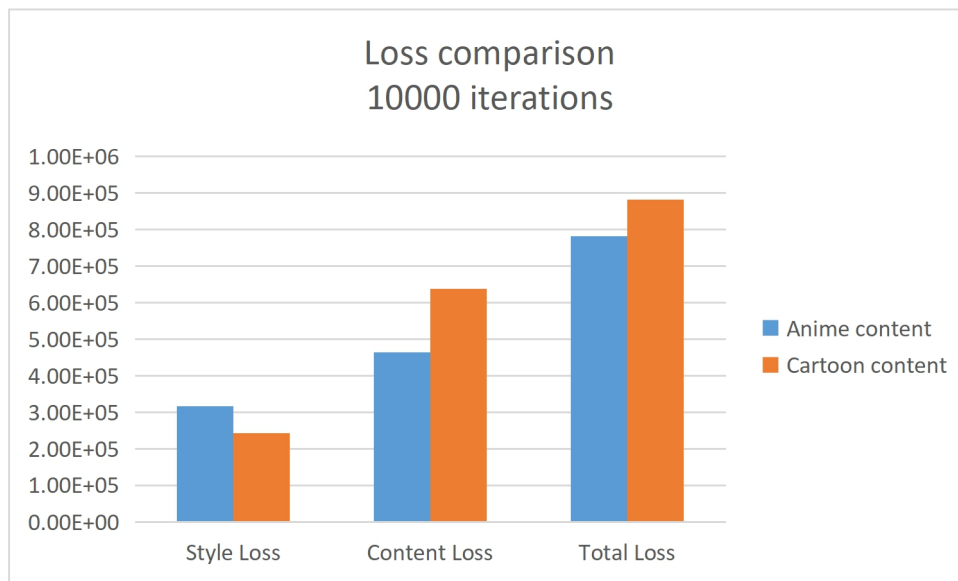
Σχήμα 44: Αποτέλεσμα μεταφοράς στυλ (Πείραμα 4)

4.2.4.1 Σύγκριση αποτελεσμάτων

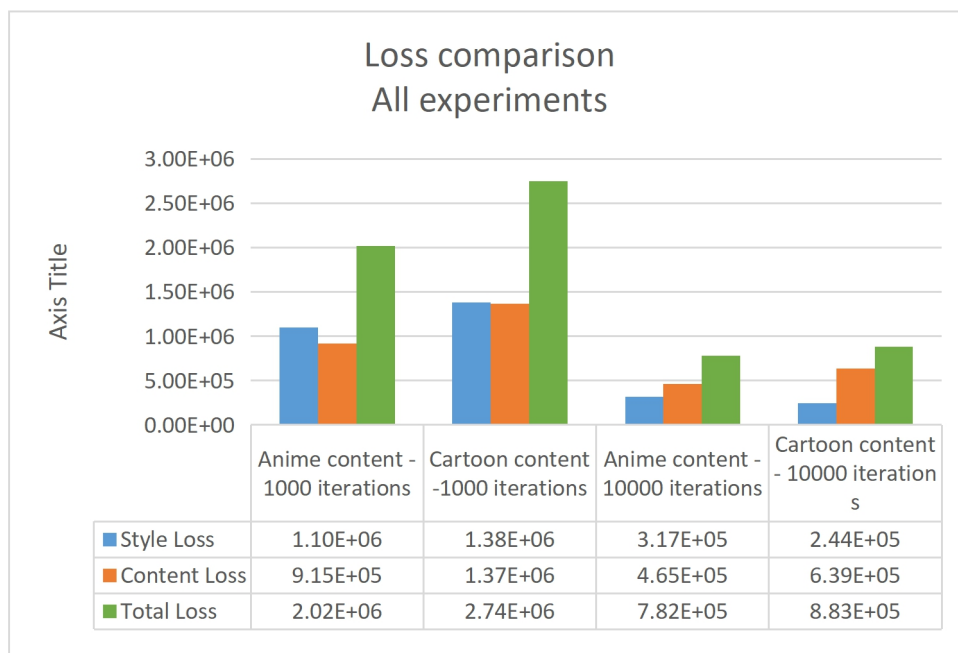
Παρακάτω παρουσιάζονται διαγράμματα για την σύγκριση των αποτελεσμάτων των πειραμάτων μεταφοράς στυλ.



Σχήμα 45: Διάγραμμα σύγκρισης αποτελεσμάτων μεταφοράς στυλ 1000 επαναλήψεων (Πειράματα 1 και 2)



Σχήμα 46: Διάγραμμα σύγκρισης αποτελεσμάτων μεταφοράς στυλ 10000 επαναλήψεων (Πειράματα 3 και 4)



Σχήμα 47: Συγκεντρωτικό διάγραμμα αποτελεσμάτων μεταφοράς στυλ

Είναι ξεκάθαρο, τόσο από τις τελικές εικόνες, όσο και από τις μετρικές απώλειας, πως στο συγκεκριμένο πείραμα, η μεταφορά του στυλ της εικόνας που ανήκει στην κατηγορία "cartoon" στην εικόνα περιεχομένου που ανήκει στην κατηγορία "anime" βγάζει καλύτερα αποτελέσματα, τόσο στα πειράματα με 1000 επαναλήψεις, όσο και σε αυτά με 10000 επαναλήψεις.

Παρατηρούμε επίσης πως ενώ στο πείραμα με τις 10000 επαναλήψεις το στυλ της εικόνας στυλ φαίνεται να έχει μεταφερθεί περισσότερο στην εικόνα περιεχομένου, σε σχέση με τα πειράματα με 1000 επαναλήψεις, με αποτέλεσμα οι μετρικές απώλειας να είναι σημαντικά πιο χαμηλές, ποιοτικά τα αποτελέσματα δεν είναι απαραίτητα καλύτερα στο μάτι.

Κεφάλαιο 5 - Συμπεράσματα και Βελτιώσεις

Στην παρούσα διπλωματική εργασία εξετάσαμε τα προβλήματα της αναγνώρισης, ταξινόμησης και μεταφοράς στυλ. Για την αναγνώριση και ταξινόμηση στυλ παρουσιάστηκαν 3 μοντέλα, τα οποία εκπαιδεύτηκαν στο ίδιο σύνολο δεδομένων, και στην συνέχεια αξιολογήθηκαν και συγκρίθηκαν τα αποτελέσματά τους. Για την μεταφορά στυλ δημιουργήθηκε ένα μοντέλο το οποίο πέτυχε σε σημαντικό βαθμό την μεταφορά στυλ από την μια εικόνα στην άλλη και αντίστροφα.

Και τα δυο μέρη παρουσιάζουν σχετικά καλά αποτελέσματα, όμως επιδέχονται πολλές βελτιώσεις που θα μπορούσαν να γίνουν στο μέλλον. Όπως αναφέρθηκε και νωρίτερα, ο κυριότερος λόγος που τα αποτελέσματα δεν ήταν καλύτερα είναι η έλλειψη υπολογιστικών πόρων.

Τα μοντέλα που χρησιμοποιήθηκαν για την αναγνώριση στυλ συνάντησαν σε έναν βαθμό το πρόβλημα της υπερεκπαίδευσης, το οποίο θα μπορούσε να αντιμετωπιστεί με ένα μεγαλύτερο ή πιο χαρακτηριστικό σύνολο δεδομένων ή με την καλύτερη επιλογή των παραμέτρων και των τελευταίων στρωμάτων των δικτύων.

Για την μεταφορά στυλ, η αξιολόγηση των αποτελεσμάτων είναι συνήθως υποκειμενική, αλλά τα αποτελέσματα θα μπορούσαν να ήταν καλύτερα είτε με την αφιέρωση περισσότερου χρόνου και επαναλήψεων, είτε με την χρήση διαφορετικών τεχνικών μεταφοράς στυλ, όπως είναι για παράδειγμα η χρήση GAN.

Βιβλιογραφία

- Advani, V., 2021. What is Machine Learning | Definition, Tools, how it Works & Uses. [online] GreatLearning Blog: Free Resources what Matters to shape your Career!. Available at: <<https://www.mygreatlearning.com/blog/what-is-machine-learning/>> [Accessed 6 June 2021].
- Altman, N., 1992. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. The American Statistician, 46(3), pp.175-185. [doi:10.1080/00031305.1992.10475879](https://doi.org/10.1080/00031305.1992.10475879)
- Berezhnoy, I., Postma, E. and van den Herik, J., 2007. Computer analysis of Van Gogh's complementary colours. Pattern Recognition Letters, 28(6), pp.703-709. <https://doi.org/10.1016/j.patrec.2006.08.002>
- Brownlee, J., 2017. A Gentle Introduction to Transfer Learning for Deep Learning. [online] Machine Learning Mastery. Available at: <<https://machinelearningmastery.com/transfer-learning-for-deep-learning/>> [Accessed 6 June 2021].
- Brownlee, J., 2021. How to Choose an Activation Function for Deep Learning. [online] Machine Learning Mastery. Available at: <<https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/>> [Accessed 5 June 2021].
- Castañón, J., 2019. 10 Machine Learning Methods that Every Data Scientist Should Know. [online] Medium. Available at: <<https://towardsdatascience.com/10-machine-learning-methods-that-every-data-scientist-should-know-3cc96e0e0009>> [Accessed 6 June 2021].
- Chandra, A., 2018. McCulloch-Pitts Neuron — Mankind's First Mathematical Model Of A Biological Neuron. [online] Medium. Available at: <<https://towardsdatascience.com/mcculloch-pitts-model-5fdf65ac5dd1>> [Accessed 5 June 2021].
- Chaurasia, A., 2020. Explore Neural Style Transfer with Weights & Biases. [online] Medium. Available at: <<https://medium.com/analytics-vidhya/explore-neural-style-transfer-with-weights-biases-344533d7b080>> [Accessed 5 June 2021].
- Dumoulin, V., Shlens, J. and Kudlur, M., 2016. A Learned Representation For Artistic Style. [arXiv:1610.07629](https://arxiv.org/abs/1610.07629).
- Foot, K., 2019. A Brief History of Machine Learning - DATAVERSITY. [online] DATAVERSITY. Available at: <<https://www.dataversity.net/a-brief-history-of-machine-learning/>> [Accessed 5 June 2021].
- Gatys, L., Ecker, A. and Bethge, M., 2015. A Neural Algorithm of Artistic Style. [arXiv:1508.06576](https://arxiv.org/abs/1508.06576).
- Huang, X. and Belongie, S., 2017. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. [arXiv:1703.06868](https://arxiv.org/abs/1703.06868).
- Ioffe, S. and Szegedy, C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. [arXiv:1502.03167](https://arxiv.org/abs/1502.03167).
- Jiang, S., Shao, M., Xiong, C. and Fu, Y., 2019. Style Recognition and Kinship Understanding. Deep Learning Through Sparse and Low-Rank Modeling, pp.213-249. <https://doi.org/10.1016/B978-0-12-813659-1.00009-3>.

Johnson, C., Hendriks, E., Berezhnoy, I., Brevdo, E., Hughes, S., Daubechies, I., Li, J., Postma, E. and Wang, J., 2008. Image processing for artist identification. *IEEE Signal Processing Magazine*, 25(4), pp.37-48. DOI: [10.1109/MSP.2008.923513](https://doi.org/10.1109/MSP.2008.923513)

Johnson, J., 2019. Towards the Algorithmic Detection of Artistic Style. *International Journal of Advanced Computer Science and Applications(ijacsa)*, 10(1).
<http://dx.doi.org/10.14569/IJACSA.2019.0100109>

Johnson, J., Alahi, A. and Fei-Fei, L., 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. <https://arxiv.org/abs/1603.08155>

Le, J., 2018. The 4 Convolutional Neural Network Models That Can Classify Your Fashion Images. [online] Medium. Available at: <<https://towardsdatascience.com/the-4-convolutional-neural-network-models-that-can-classify-your-fashion-images-9fe7f3e5399d>> [Accessed 5 June 2021].

Lee, S. and Cha, E., 2016. Style classification and visualization of art painting's genre using self-organizing maps. *Human-centric Computing and Information Sciences*, 6(1).
<https://doi.org/10.1186/s13673-016-0063-4>.

Li, J., Yao, L., Hendriks, E. and Wang, J., 2012. Rhythmic Brushstrokes Distinguish van Gogh from His Contemporaries: Findings via Automated Brushstroke Extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6), pp.1159-1176. DOI: [10.1109/TPAMI.2011.203](https://doi.org/10.1109/TPAMI.2011.203)

Lombardi, T., 2005. The classification of style in fine-art painting.
<https://digitalcommons.pace.edu/dissertations/AAI3189084/>

McCulloch, W. and Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), pp.115-133. <https://doi.org/10.1007/BF02478259>.

MK Gurucharan, 2020. Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network | upGrad blog. [online] upGrad blog. Available at: <<https://www.upgrad.com/blog/basic-cnn-architecture/>> [Accessed 5 June 2021].

Neurohive.io. 2021. VGG16 - Convolutional Network for Classification and Detection. [online] Available at: <<https://neurohive.io/en/popular-networks/vgg16/>> [Accessed 5 June 2021].

Pan, S. and Yang, Q., 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), pp.1345-1359. <https://ieeexplore.ieee.org/document/5288526>

PHANI8, 2020. Convolutional Neural Network Architecture | CNN Architecture. [online] Analytics Vidhya. Available at: <<https://www.analyticsvidhya.com/blog/2020/10/what-is-the-convolutional-neural-network-architecture/>> [Accessed 5 June 2021].

Rigau, J., Feixas, M. and Sbert, M., 2008. Informational Dialogue with Van Gogh's Paintings. [online] Dx.doi.org. Available at: <<http://dx.doi.org/10.2312/COMPAESTH/COMPAESTH08/115-122>> [Accessed 11 June 2021].

Rosenblatt, F., 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), pp.386-408. <https://doi.org/10.1037/h0042519>.

Sas.com. 2021. Machine Learning: What it is and why it matters. [online] Available at: <https://www.sas.com/en_us/insights/analytics/machine-learning.html> [Accessed 6 June 2021].

Simonyan, K. and Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. [arXiv 1409.1556](https://arxiv.org/abs/1409.1556).

Surma, G., 2019. Style Transfer - Styling Images with Convolutional Neural Networks. [online] Medium. Available at: <<https://gsurma.medium.com/style-transfer-styling-images-with-convolutional-neural-networks-7d215b58f461>> [Accessed 5 June 2021].

Valueva, M., Nagornov, N., Lyakhov, P., Valuev, G. and Chervyakov, N., 1990. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. Mathematics and Computers in Simulation, 177, pp.232-243.
[doi:10.1016/j.matcom.2020.04.031](https://doi.org/10.1016/j.matcom.2020.04.031)

Vieira, V., Fabbri, R., Sbrissa, D., da Fontoura Costa, L. and Travieso, G., 2015. A quantitative approach to painting styles. Physica A: Statistical Mechanics and its Applications, 417, pp.110-129.
<https://doi.org/10.1016/j.physa.2014.09.038>

Yosinski, J., Clune, J., Bengio, Y. and Lipson, H., 2014. How transferable are features in deep neural networks?. <https://arxiv.org/abs/1411.1792>

Yuan, R., 2018. Neural Style Transfer: Creating Art with Deep Learning using tf.keras and eager execution. [online] Medium. Available at: <<https://medium.com/tensorflow/neural-style-transfer-creating-art-with-deep-learning-using-tf-keras-and-eager-execution-7d541ac31398>> [Accessed 5 June 2021].

Zhang, W., Itoh, K., Tanida, J. and Ichioka, Y., 1990. Parallel distributed processing model with local space-invariant interconnections and its optical architecture. Applied Optics, 29(32), p.4790.
[doi:10.1364/AO.29.004790](https://doi.org/10.1364/AO.29.004790)

Παράρτημα

Ευρετήριο Ακρωνυμίων και Συντμήσεων

CNN: Convolutional Neural Network (Συνελικτικό Νευρωνικό Δίκτυο)

FC layer: Fully Connected Layer (Πλήρως Συνδεδεμένο Στρώμα)

ReLU: Rectified Linear Unit (Διορθωμένη Γραμμική Μονάδα)

TP: True Positive (Αληθώς Θετικό)

FP: False Positive (Ψευδώς Θετικό)

TN: True Negative (Αληθώς Αρνητικό)

FN: False Negative (Ψευδώς Αρνητικό)

NLP: Natural Language Processing (Επεξεργασία Φυσικής Γλώσσας)

GAN: Generative Adversarial Network (Γεννητικό Ανταγωνιστικό Δίκτυο)

NST: Neural Style Transfer (Μεταφορά στυλ)