



Διπλωματική Εργασία

**ΒΙΟΜΗΧΑΝΙΚΕΣ ΕΦΑΡΜΟΓΕΣ ΤΗΣ
ΒΑΘΙΑΣ ΜΑΘΗΣΗΣ**

από

ΝΙΚΟ ΓΚΑΡΡΟ

ΑΜ:222017005

Διατριβή για λήψη Βασικού Διπλώματος

Τμήμα Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής

Πανεπιστήμιο Δυτικής Αττικής

Επιβλέπων Καθηγητής

ΝΙΚΟΛΑΟΥ ΓΡΗΓΟΡΗΣ

Αθήνα, Σεπτέμβριο 2023

Βιομηχανικές Εφαρμογές της Βαθιάς Μάθησης
Μέλη Εξεταστικής Επιτροπής συμπεριλαμβανομένου και του Εισηγητή

Η διπλωματική εργασία εξετάστηκε επιτυχώς από την κάτωθι Εξεταστική Επιτροπή:

A/α	ΟΝΟΜΑ ΕΠΩΝΥΜΟ	ΒΑΘΜΙΔΑ/ΙΔΙΟΤΗΤΑ	ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ
	Νικολάου Γρηγόριος		
	Βασιλειάδου Σουλτάνα		
	Δρόσος Χρήστος		

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος ΝΙΚΟ ΓΚΑΡΡΟ, με αριθμό μητρώου 222017005 φοιτητής του Πανεπιστημίου Δυτικής Αττικής του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της 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

ΝΙΚΟ ΓΚΑΡΡΟ



Πίνακας Περιεχομένων

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ	4
Πίνακας Περιεχομένων	6
Κατάλογος εύρεσης Εικόνων	8
Κατάλογος εύρεσης Πινάκων	11
Περίληψη	12
Λέξεις κλειδιά	12
Abstract	13
Keywords	13
Αναγνωρίσεις	14
Κεφάλαιο 1 - Εφαρμογές της Υπολογιστικής Όρασης στην Βιομηχανική Παραγωγή	15
1.1 - Περίληψη κεφαλαίου	15
1.2.1 - Προληπτική Συντήρηση	15
1.2.2 - 3D Επιθεώρηση	18
1.2.3 - Επιθεώρηση συσκευασίας	19
1.2.4 - Αυτόματη ταξινόμηση και μέτρηση	19
1.2.5 - Παρακολούθηση συμμόρφωσης	19
1.2.6 - Εφοδιαστική αλυσίδα	20
1.2.7 - Πρόβλεψη αποθέματος	21
1.2.8 - Ρομποτική και αυτοματισμός	22
1.2.9 - Διασφάλιση ποιότητας	22
1.3 - Εισαγωγή στη Μηχανική Όραση	23
1.3.1 - Λειτουργία ενός συστήματος υπολογιστικής όρασης	24
1.3.2 - Τα συστήματα ορασης βελτιώνουν την ποιότητα των προϊόντων	26
1.3.3 - Βιομηχανίες που χρησιμοποιούν μηχανική όραση	26
1.3.4 - Τα συστήματα όρασης είναι σημαντικά για την φαρμακοβιομηχανία	27
2. Αύξηση της παραγωγικότητας	28
3. Μειώνονται τα έξοδα	28
1.3.5 - Προκλήσεις για την υιοθέτηση της επιθεώρηση με μηχανική όραση στην πραγματικότητα	29
1.3.6 - Το μέλλον για τα συστήματα όρασης	31
1.3.7 - Συμπέρασμα	32
Κεφάλαιο 2 - Βαθιά Νευρωνικά Δίκτυα	34
2.1 - Περίληψη	34
2.2 - Το perceptron	34
2.3 - Πλήρως Συνδεδεμένα Πυκνά Στρώματα (Fully Connected Dense Layer)	35
2.2 - Συνελκτικό Νευρωνικό Δίκτυο	37
2.2.1 - Επίπεδο συνέλιξης	37

2.2.2 - Padding.....	38
2.2.3 - Διασκελισμός (Stride)	39
2.2.4 - Στρώμα υποδειγματοληψίας (pooling layer).....	40
2.2.5 - Μη γραμμική συνάρτηση ενεργοποίησης νευρώνα (ReLU)	41
Κεφάλαιο 3 – Εφαρμογή της βαθιας μαθησης στην κατηγοριοποίηση	
ελλωματικών χαπιών στην φαρμακοβιομηχανία	42
3.1 – Περίληψη εφαρμογής	42
3.2 – Λόγος Εφαρμογής	42
3.2 – Τα δεδομένα	43
3.3 – Τα μοντέλα	43
3.4 - Διαδικασία κατασκευής	45
3.4.1 - Προετοιμασία Δεδομένων	45
3.4.2 - Μοντέλο	53
3.5 - Εκπαίδευση μοντέλου	57
3.5.1 Fine Tuning	60
3.6 - Εξαγωγή συμπερασμάτων.....	63
3.6.1 - Στατιστικά στοιχεία.....	65
3.7 - Εκπαίδευση με άλλα δημοφιλή μοντέλα και σύγκριση για την εύρεση του καλύτερου.	69
3.7.1 Fine Tuning	81
3.7.2 - Εξαγωγή συμπερασμάτων	89
3.7.3 - Στατιστικά στοιχεία	95
3.8 - Εξαγωγή μοντέλων	104
Συμπεράσματα Εφαρμογής.....	107
Συμπεράσματα	108
Παράρτημα 1.....	109
Βιβλιογραφία.....	113

Κατάλογος εύρεσης Εικόνων

Εικόνα 1.1: Εφαρμογή της προβλεπτικής συντήρησης μέσω της ανάλυσης σημάτων.	14
Εικόνα 1.2: Τρισδιάστατη επιθεώρηση κινητήρα.	17
Εικόνα 1.3: Επιθεώρηση συσκευασίας στην βιομηχανία τροφίμων.	18
Εικόνα 1.4: Χρήση μοντέλου ανίχνευσης αντικειμένων για την επίβλεψη συμμόρφωσης.	19
Εικόνα 1.5: Σάρωση γραμμοκώδικα για την ιχνηλάτηση πακέτων.	20
Εικόνα 1.6: Χρήση μηχανικής όρασης για την ενίσχυση ρομποτικών εφαρμογών.	21
Εικόνα 1.7: (α) Grayscale. (β) RGB. (γ) Κόκκινη Λαμπρότητα . (δ) Πράσινη Λαμπρότητα . (ε) Μπλέ Λαμπρότητα .	23
Εικόνα 1.8: Πλαίσιο συστήματος υπολογιστικής όρασης για την βιομηχανική παραγωγή.	24
Εικόνα 1.9: (α) Το ιδανικό. (β) Η πραγματικότητα.	29
Εικόνα 1.10: Η εικόνα 1.7 περασμένο με τον canny edge detector.	31
Εικόνα 2.1: Το perceptron.	34
Εικόνα 2.2: Διαγραμματική αντιπροσωπεία του perceptron. Με γραμμές συμβολίζονται τα βάρη ενώ με κύκλους τους νευρώνες (nodes perceptron).	35
Εικόνα 2.3: Πολλαπλασιασμός ανά στοιχείο και άθροιση των αποτελεσμάτων στον χάρτη χαρακτηριστικών σε συνελκτικό επίπεδο.	37
Εικόνα 2.4: Αποτέλεσμα του padding "same" η διάσταση του αρχικού πίνακα δεν μεταβάλλεται.	38
Εικόνα 2.5: Το βήμα με το οποίο "κινείται" το φίλτρο πάνω στον πίνακα χαρακτηριστικών.	39
Εικόνα 2.6: (α) Υποδειματοληψία με μέγιστης τιμής. (β) Υποδειματοληψία με μέση τιμή.	40

Εικόνα 2.7: ReLU συνάρτηση ενεργοποίησης.	40
Εικόνα 3.1: Κώδικας για κατέβασμα και εξαγωγή των δεδομένων.	45
Εικόνα 3.2: Διαδρομή των αρχείων.	45
Εικόνα 3.3: Κώδικας για την μετονομασία των εικόνων.	46
Εικόνα 3.4: Κώδικας για την δημιουργία των φακέλων και των διαχωρισμό των δεδομένων σε αυτούς.	47
Εικόνα 3.5: Κώδικας για την διαγραφή των φακέλων.	48
Εικόνα 3.6: Τελικό directory path.	48
Εικόνα 3.7: Κώδικας για την εισαγωγή βιβλιοθηκών και χρήση του image_dataset_from_directory.	49
Εικόνα 3.8: Κώδικας για την γραφικοποιηση των πρώτων 9 εικόνων από τα δεδομένα ελέγχου μαζί με τα label τους.	49
Εικόνα 3.9: Γραφική αναπαράσταση των 9 πρώτων δεδομένων εκπαίδευσης.	50
Εικόνα 3.10: Κώδικας για τη προετοιμασία των εικόνων.	51
Εικόνα 3.11: Κωδικας για την γραφικοποίηση μιας εικόνας περασμένη πολλές φορές από το data_augmentaiion στρώμα.	52
Εικόνα 3.12: Γραφική αναπαράσταση μιας εικόνας από το augmentation layer.	52
Εικόνα 3.13: Η τελική αρχιτεκτονική του μοντέλου βασισμένο το mobilenetv2 μοντέλο.	54
Εικόνα 3.14: Γραφική αναπαράσταση της αρχιτεκτονικής του μοντέλου MobileNet2	56
Εικόνα 3.15: Εξίσωση σφάλματος binary cross-entropy.	57
Εικόνα 3.16: Κώδικας για την κατασκευή του custom callback και compile του μοντέλου με τις επιθυμητές υπερπαραμέτρους.	58
Εικόνα 3.17: Κώδικας αρχικοποίηση υπερ-παραμέτρων.	58
Εικόνα 3.18: Κώδικας γραφικοποίηση της ακρίβειας και του σφάλματος για τα δεδομένα ελέγχου και εκπαίδευση ανά εποχή.	59
Εικόνα 3.19: Διάγραμμα ακρίβειας, σφάλματος- εποχής για τα δεδομένα εκπαίδευσης και ελέγχου για το mobilenetv2.	60
Εικόνα 3.20: Κώδικας για την γραφικόποίηση των αποτελεσμάτων πριν και μετά του fine tuning.	62

Εικόνα 3.21: Γραφική αναπαράσταση της ακρίβειας και του σφάλματος για τα δεδομένα εκπαίδευσης και ελέγχου ανά εποχή πριν και μετά το fine tuning για το μοντέλο MobileNetv2.	63
Εικόνα 3.22: Κώδικας για τον έλεγχο της ικανότητας πρόβλεψης σωστών κατηγοριών του μοντέλου σε ένα batch των δεδομένων ελέγχου.	64
Εικόνα 3.23: Γραφική απεικόνιση 9 εικόνων από το σύνολο ελέγχου.	65
Εικόνα 3.24: Μήτρα σύγκρισης για mobilenetv2.	68
Εικόνα 3.25: Ο χάρτης χαρακτηριστικών για το μοντέλο α) VGG16, β) Resnet50, γ) Inceptionv3, δ) EfficientNetB0, ε) VGG19	70
Εικόνα 3.26: Γραφική απεικόνιση της αρχιτεκτονικής του μοντέλου με βασικό μοντέλο α) VGG16, β)Resnet50, γ)Inceptionv3, δ) EfficientNetB0, ε) VGG19 με το μέγεθος των εισόδων και των εξόδων του σε κάθε στρώμα.	72
Εικόνα 3.27: Γραφική αναπαράσταση της ακρίβειας και του σφάλματος για τα δεδομένα εκπαίδευσης και ελέγχου ανά εποχή για το μοντέλο α) VGG16, β) Resnet50, γ) Inceptionv3, δ) EfficientNetB0, ε) VGG19 .	78
Εικόνα 3.28: Γραφική αναπαράσταση της ακρίβειας και του σφάλματος για τα δεδομένα εκπαίδευσης και ελέγχου ανά εποχή πριν και μετά το fine tuning για το μοντέλο α) VGG16, β) Resnet50, γ)Inceptionv3, δ) EfficientNetB0, ε) VGG19.	87
Εικόνα 3.29: Γραφική αναπαράσταση προβλεπομένων και πραγματικών κατηγοριών για 9 εικόνες από δεδομένα ελέγχου για α) VGG16, β) Resnet50, γ)Inceptionv3, δ) EfficientNetB0, ε) VGG19.	94
Εικόνα 3.30: Μήτρα σύγκρισης για το μοντέλο α) VGG16, β) Resnet50, γ) Inceptionv3, δ) EfficientNetB0, ε) VGG19.	96
Εικόνα 3.31. α) Block 1- convolution 2. β) Block 2- convolution 2. γ) Block 3- convolution 3. δ) Block 4- convolution 3. ε) Block 5- convolution 3.	104

Κατάλογος εύρεσης Πινάκων

Πίνακας 3.1: Στατιστικά στοιχεία ταξινόμησης για mobilenetv2.	68
Πίνακας 3.2: Στατιστικά στοιχεία ταξινόμησης για α) VGG16, β) Resnet50, γ) Inceptionv3, δ) EfficientNetB0, ε) VGG19.	96
Πίνακας 3.3: Στατιστικά των μοντέλων που χρησιμοποιήθηκαν σαν βάση για την εφαρμογή.	106

Περίληψη

Τα τελευταία χρόνια, η σύγκλιση προηγμένων υπολογιστικών δυνατοτήτων και τεράστιων ποσοτήτων δεδομένων έχει οδηγήσει σε μια αλλαγή τεχνολογικού τοπίου με εκτεταμένες επιπτώσεις σε όλους τους κλάδους. Στην πρώτη γραμμή αυτού του μετασχηματισμού βρίσκεται η βαθιά μάθηση, ένα υποσύνολο τεχνητής νοημοσύνης που έχει επιδείξει αξιοσημείωτη ικανότητα στην αποκρυπτογράφηση περίπλοκων μοτίβων και στη δημιουργία μοντέλων από πολύπλοκα σύνολα δεδομένων.

Αυτή η διατριβή εμβαθύνει στις δυνατότητες της βαθιάς μάθησης σε διάφορους βιομηχανικούς τομείς και ειδικότερα στην εφαρμογή της στον ποιοτικό έλεγχο στην φαρμακοβιομηχανία. Καθώς η συνένωση δεδομένων παραγωγής και αλγορίθμων βαθιάς μάθησης έχει αναδειχθεί ως ένας κρίσιμος τεχνολογικός παράγοντας σε όλους τους κλάδους παραγωγής, αυτή η έρευνα διερευνά τις πολύπλευρες εφαρμογές της βαθιάς μάθησης, στον ποιοτικό έλεγχο, τη βελτιστοποίηση της αλυσίδας εφοδιασμού, την προγνωστική συντήρηση και τις διαδικασίες λήψης αποφάσεων. Σκοπός της διπλωματικής εργασίας είναι η κατασκευή και εφαρμογή ενός μοντέλου βαθιάς μάθησης με προσιτό και ευέλικτο τρόπο για τον ποιοτικό έλεγχο φαρμακευτικών προϊόντων μέσω συστήματος υπολογιστικής όρασης.

Λέξεις κλειδιά

Βιομηχανία, Παραγωγή, Έλεγχος ποιότητας, Επιθεώρηση, Μηχανική Όραση, Βαθιά Μάθηση, Βαθιά Νευρωνικά Δίκτυα, Υπολογιστική Όραση, Συνελκτικό Στρώμα, VGG16, ResNet50, InceptionV3, EfficientNetB0, VGG19, MobileNetV2, Φαρμακοβιομηχανία.

Abstract

In recent years, the convergence of advanced computing capabilities and massive amounts of data has led to a changing technological landscape with far-reaching implications across all industries. At the forefront of this transformation is deep learning, a subset of artificial intelligence that has demonstrated remarkable ability to decipher complex patterns and build models from complex datasets.

This thesis delves into the possibilities of deep learning in various industrial sectors and in particular its application to quality control in the pharmaceutical industry. As the fusion of production data and deep learning algorithms has emerged as a critical technological factor in all industries, this research explores the multifaceted applications of deep learning in quality control, supply chain optimization, predictive maintenance, and decision-making processes. The purpose of this dissertation thesis is to build and apply a deep learning model in an accessible and flexible way to quality control pharmaceutical products through a computer vision system.

Keywords

Industry, Manufacturing, Quality Control, Inspection, Machine Vision, Deep Learning, Deep Neural Networks, Computer Vision, Convolution Layer, VGG16, ResNet50, InceptionV3, EfficientNetB0, VGG19, MobileNetV2, Pharmaceutical Industry.

Αναγνωρίσεις

Αρχικά, θα ήθελα να εκφράσω τις ευχαριστίες μου στον κύριο επιβλέποντα Καθηγητή Γρηγόρη Νικολάου για τη συνεχή υποστήριξη της διπλωματικής μου διατριβής και για την υπομονή καθώς και για τα κίνητρα και τις ιδέες που μου έδωσε.

Από τις πρώτες συζητήσεις, ο Καθηγητής Γρηγόρης Νικολάου με βοήθησε να καταλάβω ποια βήματα πρέπει να ακολουθήσω για να αποκτήσω τις απαραίτητες γνώσης για να ολοκληρώσω με επιτυχία τη διατριβή μου.

Θα ήθελα επίσης να εκφράσω τη βαθύτατη εκτίμηση μου στους συναδέλφους μου που διάβασαν και σχολίασαν τη διατριβή μου.

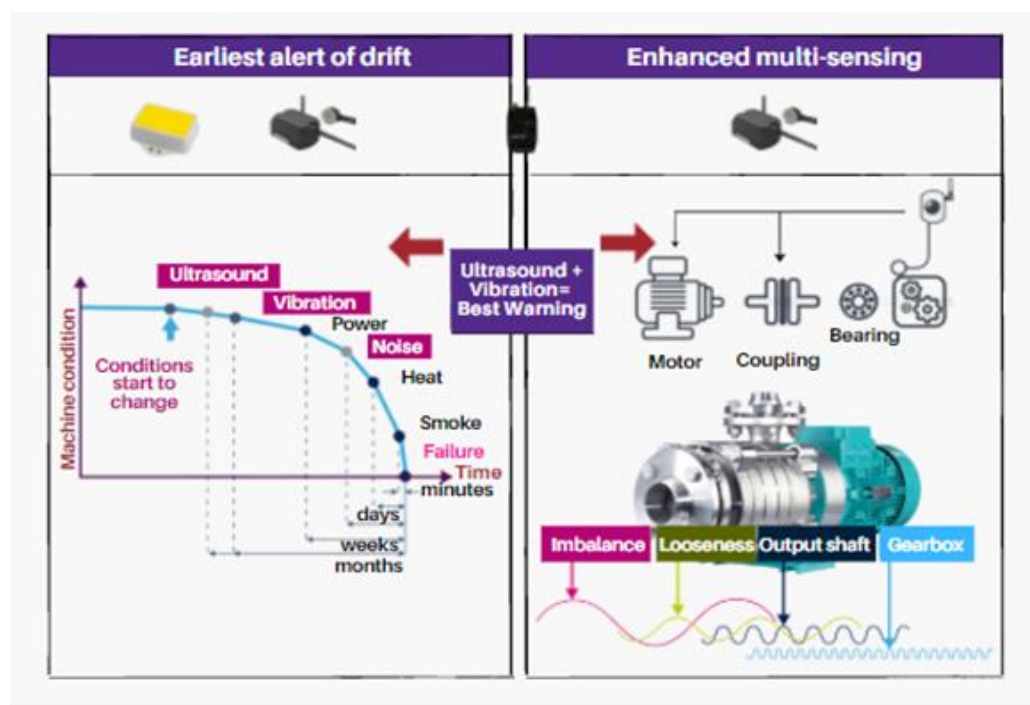
Ευχαριστώ την οικογένεια μου για την για την ενθάρρυνση που έδειξε όλο αυτό το διάστημα σε όσες δυσκολίες υγείας που είχα αντιμετώπιση κατά την συγγραφή της εργασίας.

Κεφάλαιο 1 - Εφαρμογές της Υπολογιστικής Ορασης στην Βιομηχανική Παραγωγή

1.1 - Περίληψη κεφαλαίου

Αυτό το κεφάλαιο εισάγει τις κυριότερες εφαρμογές της βαθιάς μάθησης στην βιομηχανία. Στόχος του κεφαλαίου είναι η περιγραφή των έως τώρα γνωστών εφαρμογών και τις σημασίες τους στην γραμμή παραγωγής. Επίσης θα αναλυθούν με μεγαλύτερη λεπτομέρεια η εφαρμογές της βαθιάς μάθησης και άλλων παραδοσιακών τεχνικών ποιοτικού ελέγχου όπως η μηχανική όραση στην φαρμακοβιομηχανία το οποίο θα είναι το κύριο θέμα της πρακτικής εφαρμογής, κεφάλαιο 3.

1.2.1 - Προληπτική Συντήρηση



Εικόνα 1.1. Εφαρμογή της προβλεπτικής συντήρησης μέσω της ανάλυσης σημάτων [1].

Η προληπτική συντήρηση προβλέπει τις πιθανές μελλοντικές καταστάσεις του εξοπλισμού ώστε να προσδιορίσει τις εργασίες συντήρησης. Αυτό πραγματοποιείται χρησιμοποιώντας διάφορους αισθητήρες που συλλέγουν δεδομένα και πληροφορίες από

τα διάφορα αντικείμενα και μηχανήματα κατά την διάρκεια μιας γραμμής παραγωγής, για να αποφευχθεί η βλάβη.

Η βλάβη (ή «απροσδόκητος χρόνος διακοπής λειτουργίας» όπως αναφέρεται στη βαριά βιομηχανία), σημαίνει απρογραμματίστο χρόνο διακοπής λειτουργίας, είτε από χαλασμένα μηχανήματα, καθυστερημένες προμήθειες, θέματα προσωπικού ή οποιοδήποτε είδος ζητημάτων που σχετίζονται με το εργοστάσιο. Για την αντιμετώπιση του απρογραμματίστου χρόνου διακοπής λειτουργίας, το πρώτο εργαλείο στον κατασκευαστή είναι η προγνωστική συντήρηση που είναι ιδιαίτερα ευνοϊκή για την υιοθέτηση της βαθιάς μάθησης [2].

Κυριότερος λόγος χρήσης της προληπτικής συντήρησης είναι η επιχειρηματική αξία.

Η συντήρηση σε φυσικό εξοπλισμό είναι δαπανηρή και χρονοβόρα. Εκτός από το κόστος των ανθρώπινων διαγνωστικών προσπαθειών από εκπαιδευμένους επαγγελματίες, ο χρόνος διακοπής λειτουργίας του εξοπλισμού μπορεί να έχει επιπτώσεις στην παραγωγικότητα. Όταν ένα κομμάτι του κατασκευαστικού εξοπλισμού κλείνει για συντήρηση, το απόθεμα κατά τη διαδικασία που τροφοδοτείται σε αυτό το μηχάνημα πρέπει να σταματήσει ή να δρομολογηθεί αλλού. Το εφεδρικό απόθεμα κατά τη διαδικασία μπορεί να δημιουργήσει απρόβλεπτα αποτελέσματα για άλλο εξοπλισμό, για τις μεταφορές και για το εργατικό δυναμικό που τον χειρίζεται και την πιθανή αποτυχία να τηρηθούν οι προσδοκίες και οι προθεσμίες παράδοσης των πελατών.

Η εφαρμογή της βαθιάς μάθησης στην προληπτική συντήρηση έχει ως αποτέλεσμα να μειώσει το χρόνο διακοπής λειτουργίας βελτιώνοντας παράλληλα τη συνολική συντήρηση και παρακολούθηση του εξοπλισμού.

Το κυριότερο στοιχείο κάθε συστήματος βαθιάς μάθησης είναι τα δεδομένα. Οι εφαρμογές προγνωστικής συντήρησης αξιοποιούν μια ποικιλία τύπων δεδομένων:

- Δεδομένα αισθητήρων στον ίδιο τον εξοπλισμό, όπως: αισθητήρες θερμότητας, αισθητήρες δόνησης, αισθητήρες κίνησης κ.τ.λ.
- Δεδομένα προγραμματιζόμενου λογικού ελεγκτή (PLC), που περιλαμβάνουν,

εισόδους που εισάγονται στο μηχάνημα, εξόδους από το μηχάνημα, ανθρώπινες οδηγίες για το μηχάνημα με την πάροδο του χρόνου και άλλους παράγοντες.

- Υπολογιστική όραση. Για παράδειγμα: Μια κάμερα που παρακολουθεί και ελέγχει την ποιότητα ενός προϊόντος σε μια γραμμή συναρμολόγησης.
- Δεδομένα χρονοσειρών γνωστών βλαβών (Time Series Failure Patterns), προβλημάτων ή δυσλειτουργιών, τα οποία μπορούν να συνδεθούν με δεδομένα αισθητήρα προκειμένου να προσδιοριστεί η κατάσταση του μηχανήματος που οδήγησε στο πρόβλημα.
- Χρήσιμες εξωτερικές πηγές δεδομένων, αυτό θα μπορούσε να περιλαμβάνει δεδομένα από άλλο όμοιο εξοπλισμό σε μια γραμμή παραγωγής ή ακόμη και δεδομένα καιρού (αν ο καιρός ή η εξωτερική θερμοκρασία ενδέχεται να επηρεάσουν την απόδοση του μηχανήματος).

Μετάπειτα θα πρέπει να επεξεργαστούν τα δεδομένα, αυτό πραγματοποιείται από μια ομάδα επιστημόνων δεδομένων και ειδικών συντήρησης μηχανών συνεργάζονται για να προσδιορίσουν τους παράγοντες που μπορεί να είναι πιο ενδεικτικοί για την παρακολούθηση της λειτουργίας ή δυσλειτουργίας του μηχανήματος. Διαφορετικά κομμάτια εξοπλισμού θα απαιτήσουν διαφορετικό συνδυασμό αισθητήρων και συχνά απαιτεί πειραματισμό για να προσδιοριστεί ποιος συνδυασμός δεδομένων θα είναι ο καλύτερος προγνωστικός παράγοντας βλαβών ή λειτουργίας του μηχανήματος και άρα οι καλύτεροι δείκτες συντήρησης.

Τέλος θα κατασκευαστεί ένας αλγόριθμος βαθιάς μάθησης που δώσει ένα στατιστικά επιθυμητό αποτέλεσμα πρόβλεψης. Η διαδικασία αυτή μπορεί να χρειαστεί πολλές επαναλήψεις έως ότου να επικεντρωθούν οι σημαντικότεροι παράγοντες.

Τα αποτελέσματα της προβλεπτικής συντήρησης με βαθιά μάθησης θα πρέπει να είναι:

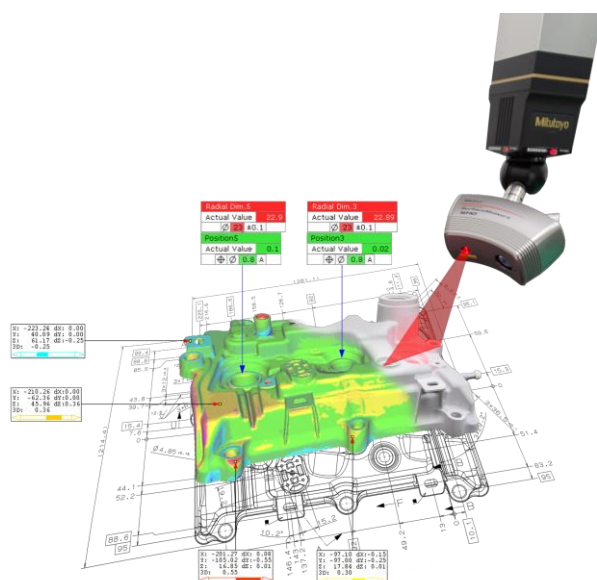
- Λιγότερες καταστροφικές αστοχίες σε δεδομένη χρονική περίοδο όγκου παραγωγής.
- Λιγότερος συνολικός χρόνος διακοπής λειτουργίας για τα εν λόγω μηχανήματα

που συντηρούνται.

- Λιγότερο χειρωνακτική διαγνωστική εργασία για ειδικούς μηχανημάτων.

1.2.2 - 3D Επιθεώρηση

Κατά την γραμμή παραγωγής τα προϊόντα περνάνε από πολλά στάδια ανίχνευσης και σάρωσης από διαφορετικές κατεύθυνσης με αποτέλεσμα να δημιουργηθούν τρισδιάστατα μοντέλα των αντικειμένων (digital twins). Αυτό μπορεί να χρησιμοποιηθεί για την εύρεση προβληματικών αντικειμένων που μπορεί να έχουν αρνητικές ή ακόμα και καταστροφικά αποτελέσματα σε μετέπειτα στάδιο της παραγωγής.



Εικόνα 1.2. Τρισδιάστατη επιθεώρηση κινητήρα [3].

1.2.3 - Επιθεώρηση συσκευασίας



Εικόνα 1.3. Επιθεώρηση συσκευασίας στην βιομηχανία τροφίμων [4].

Η συσκευασία των προϊόντων είναι ένα από τα τελικά στάδια σε μία γραμμή παραγωγής αλλά και ταυτόχρονα ένα από τα πιο κρίσιμα, για παράδειγμα σε μια φαρμακοβιομηχανία η μέτρηση των δισκίων και καψουλών που θα συσκευαστή είναι κρίσιμο στην διασφάλιση της ποιότητας, μείωση της δυσαρέσκεια των καταναλωτών και της μείωσης του επιχειρησιακού κόστους λόγω αποσύρσεων.

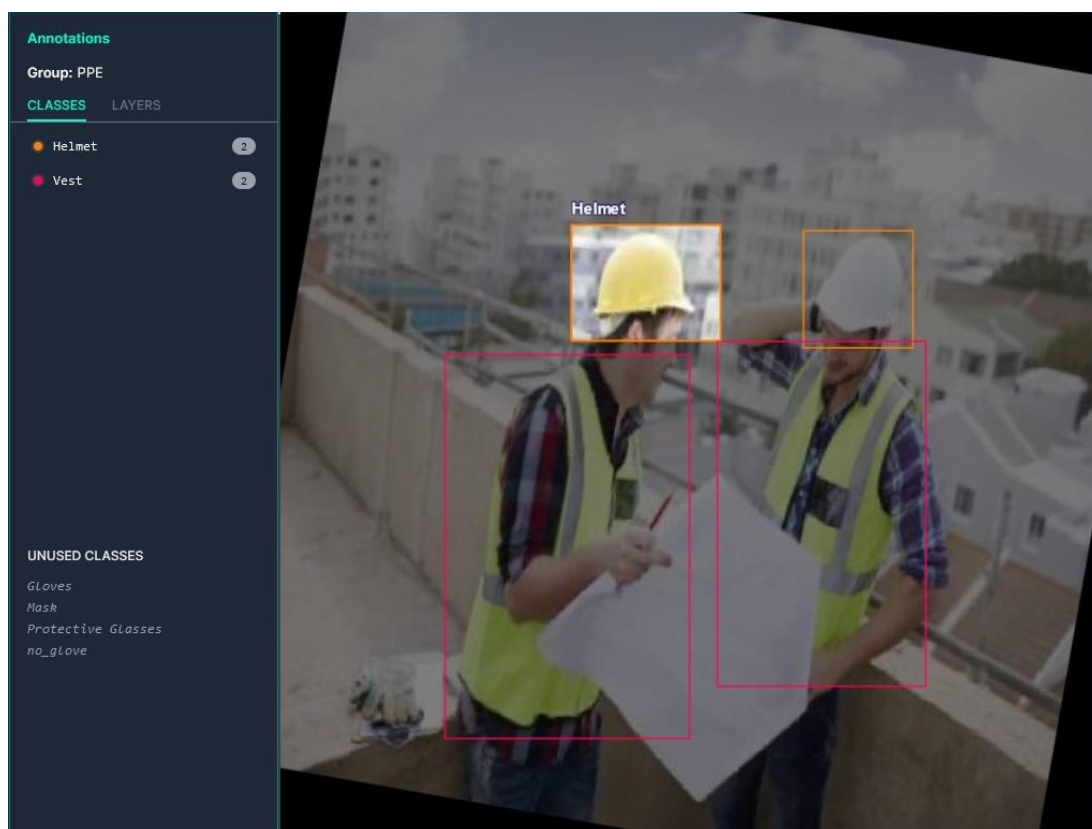
1.2.4 - Αυτόματη ταξινόμηση και μέτρηση

Με την βοήθεια της υπολογιστικής όρασης οι κατασκευαστές μπορούν χωρίς την ανθρώπινη συμβουλή αυτόματα και αυτόνομα να μετρήσουν και να ταξινομήσουν εξαρτήματα με αποτέλεσμα να αυξηθεί η ακρίβεια σε περιβάλλοντα μεγάλου όγκου παραγωγής.

1.2.5 - Παρακολούθηση συμμόρφωσης

Η υπολογιστική μάθηση με την ικανότητα της στην συνεχόμενη παρακολούθηση μπορεί να διασφαλίσει ότι η βιομηχανία ακολουθεί τους κανόνες συμμόρφωσης, δηλαδή τους κανόνες ασφάλειας για τους εργαζόμενους και τα περιβαλλοντικά πρότυπα του κάθε κράτους. Εγείρεται όμως ιδεολογικά ερωτήματα αφού η ίδια τεχνολογία μπορεί να χρησιμοποιηθεί για την παρακολούθηση των εργαζομένων στον χώρο εργασίας τους,

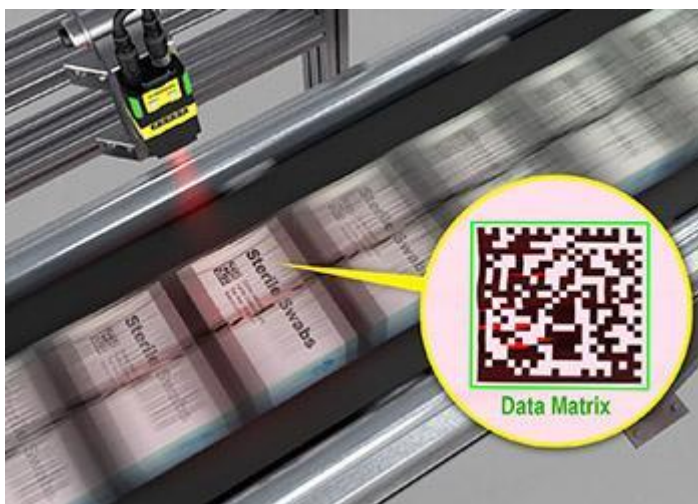
που είναι παράνομο [5].



Εικόνα 1.4. Χρήση μοντέλου ανίχνευσης αντικειμένων για την επίβλεψη συμμόρφωσης.

1.2.6 - Εφοδιαστική αλυσίδα

Με την βοήθεια αισθητηρίων και οργάνων όπως η RFID και σαρωτές γραμμοκώδικα μια βιομηχανία μπορεί να διασφαλίσει ότι πακέτα και παλέτες έχουν τους κατάλληλους γραμμοκώδικες για αναγνώριση των προϊόντων, να αντλεί δεδομένα ιχνηλατώντας την κίνηση των προϊόντων κατά την διάρκεια της εφοδιαστικής αλυσίδας. Τα δεδομένα αυτά μέσω μοντέλων μηχανικής μάθησης και άλλων παραδοσιακών μεθόδων να βελτιώσω τις διαδρομές εμπορίου και άρα την καλύτερη διαχείριση των εμπορευμάτων και την αναγνώριση και πρόβλεψη ελλείψεων.



Εικόνα 1.5. Σάρωση γραμμοκώδικα για την ιχνηλάτηση πακέτων [6].

1.2.7 - Πρόβλεψη αποθέματος

Οι παρασκευαστές αναζητούν εξωτερικά σημεία δεδομένων που μπορούν να συγκεντρώσουν – σχετικά με τους προμηθευτές τους, την αλυσίδα εφοδιασμού, τις αγορές και τα επίπεδα αποθέματός τους με την πάροδο του χρόνου – που θα τους επιτρέπουν να υπολογίζουν τα μελλοντικά επίπεδα αποθέματός.

Αυτά τα σημεία δεδομένων μπορούν να βασίζονται σε παραγγελίες υπό εξέλιξη, πωλήσεις που δεν έχουν πραγματοποιηθεί, εποχιακές διακυμάνσεις των τάσεων ζήτησης κ.τ.λ. Βλέποντας αυτές τις μεταβλητές ολιστικά και ιστορικά μέσω της ανάλυσης δεδομένων με τεχνικών βαθιάς μάθησης, οι παρασκευαστές αρχίζουν να προβλέπουν πόσο χρόνο θα χρειαστεί για να φτάσουν τα εξαρτήματα από έναν προμηθευτή με μεγαλύτερη ακρίβεια μειώνοντας τις ελλείψεις στις πρώτες ύλες και στον εξοπλισμό τους.

1.2.8 - Ρομποτική και αυτοματισμός



Εικόνα 1.6. Χρήση μηχανικής όρασης για την ενίσχυση ρομποτικών εφαρμογών [7].

Η βαθιά μάθηση ενισχύει τις δυνατότητες των βιομηχανικών ρομπότ δίνοντάς τους τη δυνατότητα να αντιλαμβάνονται και να ανταποκρίνονται στο περιβάλλον τους με δυναμικό και “έξυπνο” τρόπο. Αυτό είναι ιδιαίτερα χρήσιμο σε εργασίες που απαιτούν επιδεξιότητα και προσαρμοστικότητα δηλαδή μη στατικά περιβάλλοντα.

1.2.9 - Διασφάλιση ποιότητας

Με την βοήθεια της ανίχνευσης ανωμαλιών με βάση τη βαθιά μάθηση (deep learning based anomaly detection) και την υπολογιστική όραση (computer vision) υπάρχει η ικανότητα στους παρασκευαστές η αυτοματοποιημένη οπτική επιθεώρηση. Μπορούν δηλαδή επαναλαμβανόμενα να εξετάσουν τα προϊόντα και τα πακέτα τους προτού φύγουν από το εργοστάσιο.

Όλες οι εργασίες παρασκευής μετρώνται σε κάποιο βαθμό από την ποιότητα του προϊόντος. Αναπόφευκτα, μια ορισμένη ποσότητα προϊόντος δεν θα εγκαταλείψει τη γραμμή συναρμολόγησης στις ιδανικές του συνθήκες ή ακόμα και σε κατάσταση πωλήσεως, με τη σειρά τους, οι παρασκευαστές ψάχνουν την ποσότητα αυτή που μπορεί να περάσει από τη γραμμή συναρμολόγησης χωρίς να βλάψει τη συνολική

ποιότητα.

Εάν ένας παρασκευαστής αυτοκινήτων βλέπει στραβωμένα τιμόνια στο τέλος της γραμμής συναρμολόγησης του, τότε ένα από τα μηχανήματά του μπορεί να υπερθερμαίνεται. Όποια και αν είναι η διάγνωση, θα πρέπει να αναλυθούν η ροή δεδομένων και ιδιαίτερα τα οπτικά δεδομένα.

1.3 - Εισαγωγή στη Μηχανική Όραση

Η μηχανική όρασης είναι ένα υποπεδίο της υπολογιστικής όρασης και επιτρέπει σε μια υπολογιστική συσκευή την δυνατότητες αυτόματης λήψης, αξιολόγησης και επεξεργασίας στατικών ή κινουμένων εικόνων. Μπορεί να αυτοματοποιήσει πολλές βιομηχανικές επιθεωρήσεις, συμπεριλαμβανομένων οπτικών επιθεωρήσεων για ελαττώματα και προβλήματα, ελέγχους παρουσίας-απουσίας, επαληθεύσεις τύπου προϊόντος, μέτρα και αναγνώσεις κωδικών.

Ένα σύστημα μηχανικής όρασης χρησιμοποιεί μια κάμερα για την προβολή μιας εικόνας, οι αλγόριθμοι όρασης υπολογιστή στη συνέχεια επεξεργάζονται και ερμηνεύουν την εικόνα, πριν δώσουν εντολή σε άλλα στοιχεία του συστήματος να ενεργήσουν με βάση αυτά τα δεδομένα.

Τα πλεονεκτήματα της μηχανικής όρασης είναι από καιρό γνωστά στη βαριά βιομηχανία για σκοπούς επιθεώρησης. Οι κάμερες και οι υπολογιστές μαζί μπορούν να καταγράφουν και να επεξεργάζονται εικόνες με πολύ μεγαλύτερη ακρίβεια και ταχύτητα από οποιονδήποτε άνθρωπο. Στην εξαιρετικά ευαίσθητη κατασκευή γραμμής παραγωγής, όπως στη δημιουργία εξαρτημάτων για βηματοδότες, δεν μπορούν να γίνουν σφάλματα. Οι ανθρώπινοι επιθεωρητές είναι πολύ επικίνδυνοι για τέτοιες λεπτομερείς επιθεωρήσεις

Οι υπολογιστές δεν βλέπουν εικόνες αλλά μια μήτρα (matrix) αριθμών η τάξη της μήτρας θα εξαρτηθεί από το περιεχόμενο των εικόνων σε πληροφορίες δηλαδή μια ασπρόμαυρη (grayscale) τύπου εικόνα, όπως η παρακάτω, θα αντιπροσωπεύεται από μία μήτρα τάξης 2 δηλαδή έναν πίνακα με διαστάσεις ίσο με το μήκος και το ύψος της εικόνας (150, 150). Οι τιμές του πίνακα βρίσκονται ανάμεσα στο [0,255] με το μηδέν να αντιπροσωπεύει το μαύρο ενώ το 255 το άσπρο. Μια έγχρωμη (rgb) εικόνα θα

αντιπροσωπεύεται από μία μήτρα τάξης 3 δηλαδή τρισδιάστατο πίνακα (κύβος) με διαστάσεις ίσες με το μήκος, ύψος και βάθος της εικόνας, το βάθος στην συγκεκριμένη περίπτωση είναι ο αριθμός των χρωματικών καναλιών που είναι ίσο με 3 δηλαδή ένας πίνακας για το κοκκίνο, ένας για το πράσινο και ένας για το μπλέ.



Εικόνα 1.7. α) Grayscale. β) RGB. γ)Κόκκινη Λαμπρότητα. δ) Πράσινη Λαμπρότητα. ε) Μπλέ Λαμπρότητα.

1.3.1 - Λειτουργία ενός συστήματος υπολογιστικής όρασης

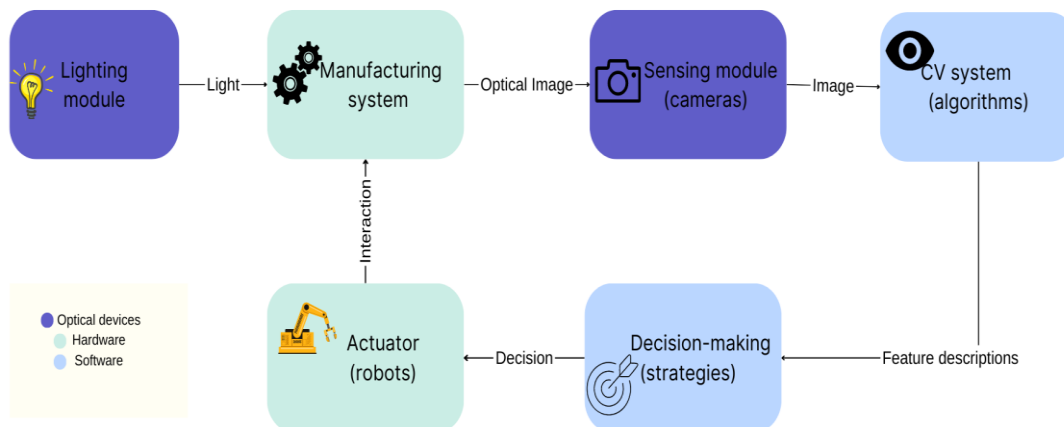
Θα περιγράφονται πώς αλληλεπιδρούν τα κυριότερα στοιχεία ελέγχου της διαδικασίας επιθεώρησης ενός προϊόντος και την χρήση της υπολογιστικής όρασης.

α) Αφού εντοπιστεί η παρουσία ενός προϊόντος από τον αισθητήρα, ξεκινά η διαδικασία, στην συνέχεια, ο αισθητήρας ενεργοποιεί μια πηγή φωτός για να φωτίσει την περιοχή του συστήματος παραγωγής και μια κάμερα για να απεικονίσει το προϊόν ή ένα από τα εξαρτήματά του.

β) Η εικόνα που τραβήχτηκε από την κάμερα μετατρέπεται σε ψηφιακά δεδομένα. Το ψηφιακό αρχείο διατηρείται σε υπολογιστή, ώστε ο αλγόριθμος της υπολογιστικής όρασης να μπορεί να το αξιολογήσει.

γ) Με τα αποτελέσματα της αξιολόγησης ένα αυτόματο σύστημα με προκαθορισμένους κανόνες να ενεργοποιήσει κατάλληλα ρομποτικά συστήματα

που θα μεταβάλλουν την γραμμή παραγωγής.



Εικόνα 1.8. Πλαίσιο συστήματος υπολογιστικής όρασης για την βιομηχανική παραγωγή [11].

- **Lighting module**

Παρέχει τη μονάδα ομοιόμορφου φωτισμού που επιτρέπει να καταγραφούν καθαρές εικόνες του συστήματος κατασκευής. **Φακός:** Οι φακοί χρησιμοποιούνται για τη συγκέντρωση του φωτός στον αισθητήρα εικόνας. **Φως:** Οποιαδήποτε διάταξη μηχανικής όρασης απαιτεί προσεκτική επιλογή φωτός. ένα σύστημα δεν μπορεί να διερευνήσει αυτό που δεν μπορεί να δει η κάμερα. Η μορφή, το μέγεθος και το χρώμα του φωτισμού, καθώς και η απόσταση και η γωνία από την οποία είναι εγκατεστημένη, μπορούν όλα να ρυθμιστούν ώστε να επισημαίνουν τα πράγματα που εξετάζονται, αποφεύγοντας ταυτόχρονα τυχόν επιπτώσεις από το περιβάλλον.

- **Manufacturing system**

Περιλαμβάνει γραμμές συναρμολόγησης, ρομπότ συναρμολόγησης και αυτοματοποιημένα κατευθυνόμενα οχήματα.

- **Sensing module**

Περιλαμβάνει κάμερες που συλλέγουν οπτικά δεδομένα από το σύστημα κατασκευής τα μετατρέπουν σε ψηφιακά δεδομένα και τα στέλνουν μέσω του Διαδικτύου των πραγμάτων (IoT) στον ελεγκτή.

- **Computer vision system**

Τυπικά τροφοδοτείται από αλγόριθμους βαθιάς μάθησης (DL), μπορεί να εκτελέσει διαφορετικές εργασίες από την καταμέτρηση προϊόντων έως τον ποιοτικό έλεγχο και ανίχνευση ελαττωμάτων (ανωμαλίας).

- **Decision-making module**

Επεξεργάζεται τα προηγούμενα αποτελέσματα βασισμένα σε κανόνες ή πιο σύνθετους αλγόριθμους τεχνητής νοημοσύνης για να ενεργοποιήσει μια κατάλληλη πορεία δράσης.

- **Actuator**

Ένα σύνολο ρομπότ που αλληλεπιδρούν με το σύστημα παραγωγής σύμφωνα με τις στρατηγικές που ορίζονται από την ενότητα λήψης αποφάσεων [12].

1.3.2 - Τα συστήματα ορασης βελτιώνουν την ποιότητα των προϊόντων

Η τεχνολογία μηχανικής όρασης επιτρέπει στον παρασκευαστή να αντικαταστήσει τη δειγματοληπτική δοκιμή με ελέγχους ποιότητας για όλα τα αντικείμενα, κάτι που πραγματοποιείται μέσω ενός συστήματος αισθητηρίων κυρίως οπτικών, κάμερες. Αυτό σημαίνει ότι κάθε παρτίδα που παράγεται μπορεί να ελεγχθεί αξιόπιστα για ελαττώματα κατά τη διαδικασία παραγωγής και χωρίς διακοπή. Ένας γρήγορος και ενδεδειγμένος έλεγχος που εγγυάται ένα ανώτερο προϊόν και μειωμένη σπατάλη.

1.3.3 - Βιομηχανίες που χρησιμοποιούν μηχανική όραση

Τα συστήματα υπολογιστικής όρασης μπορεί να είναι επωφελές για οποιαδήποτε βιομηχανική εγκατάσταση με επαναλαμβανόμενη διαδικασία. Χρησιμοποιείται ευρέως σε διάφορους τομείς, συμπεριλαμβανομένων των αυτοκινήτων, των πλαστικών, των

τροφίμων και της συσκευασίας, των ιατρικών συσκευών και των ηλεκτρονικών.

Τα συστήματα υπολογιστικής όρασης αυξάνουν την ποιότητα των προϊόντων μειώνοντας το ανθρώπινο λάθος και διασφαλίζοντας ποιοτικούς ελέγχους σε όλα τα αγαθά που διέρχονται από τη γραμμή παραγωγής. Έχει ως αποτέλεσμα, να μειώνουν το συνολικό κόστος παραγωγής τόσο σε χρόνο όσο και σε χρήμα, καθώς εμφανίζονται λιγότερα ελαττώματα και ελαττωματικά στοιχεία και δεν φτάνουν ποτέ στο επόμενο στάδιο, προκαλώντας χρονικές καθυστερήσεις. Αυτό βοηθά στην αποτροπή ελαττωματικών προϊόντων από το να φτάσουν στον τελικό πελάτη και να δημιουργήσουν δυσμενή δημοσιότητα, την οποία ορισμένες εταιρείες δεν έχουν αποφύγει.

1.3.4 - Τα συστήματα όρασης είναι σημαντικά για την φαρμακοβιομηχανία

Η φαρμακευτική βιομηχανία είναι ένας από τους πολλούς κλάδους όπου η ασφάλεια και η ποιότητα των προϊόντων είναι κρίσιμοι παράγοντες. Αυτό απαιτεί τη χρήση οπτικής επιθεώρησης στο στάδιο της παρασκευής του φαρμακευτικού προϊόντος.

Στη φαρμακευτική παραγωγή η βαθιά μάθηση χρησιμοποιείται για την επιθεώρηση της ποιότητας και της κατάστασης του προϊόντος πριν μπει στη συσκευασία του. Αυτό περιλαμβάνει εργασίες όπως η καταμέτρηση του αριθμού των χαπιών σε ένα μπουκάλι, ο έλεγχος κάθε χαπιού για το σχήμα, το μέγεθος, την ζημιά και η επικύρωση των ετικετών.

Συχνές περιπτώσεις χρήσης επιθεώρησης συσκευασίας φαρμάκων είναι:

- Επιθεώρηση των επιπέδων πλήρωσης σε φιαλίδια, μπουκάλια και άλλα δοχεία
- Έλεγχος για σωστή συναρμολόγηση και λειτουργία ιατροτεχνολογικών συσκευών και εξοπλισμού.
- Ανάγνωση και επαλήθευση κωδικών, όπως γραμμωτούς κώδικες και σειριακούς αριθμούς, για σκοπούς παρακολούθησης και ιχνηλασιμότητας.
- Ανίχνευση και αναγνώριση μικροβιολογικής ανάπτυξης σε δείγματα.

- Εξασφάλιση σωστής και ακριβούς επισήμανσης της δοσολογίας και των ημερομηνιών λήξης.
- Παρακολούθηση και έλεγχος των διαδικασιών παραγωγής για τη μείωση του κινδύνου σφαλμάτων και μόλυνσης.
- Επιθεώρηση συσκευασιών και φακελισκων με κυψέλες για να διασφαλιστεί η σωστή σφράγιση και η ακεραιότητα του μεμονωμένες δόσεις.
- Επαλήθευση της ακρίβειας των εκτυπωμένων πληροφοριών και ετικετών σε αμπούλες και σύριγγες [13].

Τα πλεονεκτήματα της εφαρμογής συστήματος όρασης με βαθιά μάθηση είναι τα εξής:

1. Τα αποτελέσματα είναι ακριβή

Τα συστήματα υπολογιστικής όρασης παρέχουν αποτελέσματα υψηλής ακρίβειας σε σύγκριση με τον χειροκίνητο έλεγχο ποιότητας προϊόντων, καθώς δεν επηρεάζονται από το ανθρώπινο λάθος. Αυτή η ακρίβεια μπορεί να βοηθήσει στη μείωση του κινδύνου πιθανών λαθών που μπορεί να οδηγήσουν σε ανακλήσεις προϊόντων ή αγωγές. Αυτά τα συστήματα είναι επίσης πιο αποτελεσματικά από άποψη χρόνου, καθώς μπορούν να επεξεργαστούν μεγάλες ποσότητες δεδομένων γρήγορα και με ακρίβεια. Βοηθά επίσης τις εταιρείες να συμβαδίζουν με τους αυστηρούς κανονισμούς που θέτουν οι κυβερνήσεις και οι ρυθμιστικοί φορείς.

2. Αύξηση της παραγωγικότητας

Τα συστήματα υπολογιστικής όρασης επιτρέπουν ταχύτερο, ακριβέστερο και ποιοτικό έλεγχο των φαρμακευτικών προϊόντων, με αποτέλεσμα μεγαλύτερη παραγωγικότητα. Αυτά τα συστήματα επιτρέπουν τη γρήγορη ανίχνευση και αναγνώριση αντικειμένων, επιτρέποντας πιο αποτελεσματικές γραμμές συσκευασίας και παραγωγής. Αυτό βοηθά στη μείωση του κόστους που σχετίζεται με τη χειρωνακτική εργασία, αυξάνοντας έτσι την απόδοση.

3. Μειώνονται τα έξοδα

Τα συστήματα υπολογιστικής όρασης παρέχουν επίσης εξοικονόμηση κόστους σε

σχέση με τις παραδοσιακές μεθόδους επιθεώρησης. Με την αυτοματοποίηση της διαδικασίας, οι εταιρείες μπορούν να μειώσουν την ανάγκη για χειρωνακτική εργασία και να εξοικονομήσουν χρήματα σε υλικά και εργασία. Επιπλέον, μπορούν να μειώσουν τον κίνδυνο δαπανηρής ζημιάς ή μόλυνσης των προϊόντων που θα μπορούσαν να προκύψουν με χειροκίνητη επιθεώρηση.

Χρησιμοποιώντας ένα σύστημα επιθεώρησης όρασης στα αρχικά στάδια παραγωγής, τα ελαττωματικά μέρη απομακρύνονται αμέσως από τη διαδικασία. Τα ελαττωματικά προϊόντα δεν συνεχίζουν ποτέ στα επόμενα στάδια κατασκευής και επομένως δεν επιβαρύνονται με περαιτέρω κόστος. Τα ελαττωματικά προϊόντα μπορούν να επανεισαχθούν στη διαδικασία παραγωγής σε μεταγενέστερο στάδιο, εξοικονομώντας έτσι υλικά. Μέσω της ανάλυσης, τα προβλήματα μπορούν να διορθωθούν στο σημείο προέλευσης, με αποτέλεσμα την αύξηση της παραγωγικότητας και της διαθεσιμότητας του συστήματος. Η τεχνολογία μηχανικής όρασης είναι μοναδική ως προς την ικανότητά της να βελτιώνει την ποιότητα, ενώ ταυτόχρονα μειώνει το κόστος και προστατεύει τη μάρκα από ανακλήσεις, πρόστιμα και δυσμενή αρνητική δημοσιότητα.

1.3.5 - Προκλήσεις για την υιοθέτηση της επιθεώρηση με μηχανική όραση στην πραγματικότητα

Η οπτική επιθεώρηση αφορά την εύρεση ελαττωματικών προϊόντων και την αφαίρεσή τους από την παραγωγική διαδικασία. Οπτική επιθεώρηση μπορεί να πραγματοποιηθεί χειροκίνητα ή αυτόματα με συστήματα υπολογιστικής όρασης. Πολλές εταιρείες χρησιμοποιούν την αυτόματη οπτική επιθεώρηση για διαφορετικούς λόγους, συνήθως για τη μείωση του κόστους, τη βελτίωση της ανίχνευσης, την αύξηση της απόδοσης παραγωγής, την κάλυψη των απαιτήσεων της αγοράς ή ακόμα και τη βελτίωση της φήμης της μάρκας [14].



Εικόνα 1.9. α) Το ιδανικό. β) Η πραγματικότητα.

Η ιδέα των συστημάτων επιθεώρησης και διαλογής είναι να ταξινομήσουν τα προϊόντα χωρίς αμφιβολία σε ομάδες σύμφωνα με τις επιθυμητές κατηγοριοποιήσεις.

Στην ιδανική περίπτωση, τα συστήματα επιθεώρησης με μηχανική όραση είναι ένας τέλειος συνδυασμός πλήρως ρυθμιζόμενου υλικού και λογισμικού που μπορούν να ρυθμιστούν ώστε να χειρίζονται όλους τους τύπους προϊόντων για να εντοπίζουν και να εξαλείφουν όλους τους τύπους οπτικών ελαττωμάτων. Στη φαρμακευτική παραγωγή υπάρχουν πολλά είδη ελαττωμάτων και κάθε προϊόν έχει τη δική του εμφάνιση με μοναδικά οπτικά ελαττώματα [14].

Παρόλο αυτά η συλλογή δεδομένων από τον φυσικό κόσμο παραμένει η μεγαλύτερη πρόκληση για τους κατασκευαστές στην υιοθέτηση της τεχνητής νοημοσύνης, επειδή είναι η πιο μοναδική για τον κάθε χρήστη [8].

Για την αντιμετώπιση των απαιτήσεων του κάθε χρήστη, τα αυτόματα συστήματα οπτικής επιθεώρησης με μηχανική όραση έχουν συχνά πολλές παραμέτρους για να τους προσδώσουν μεγάλο εύρος δυνατοτήτων, γεγονός που ενέχει υψηλό κίνδυνο μη βέλτιστης λειτουργίας. Επίσης θα πρέπει να σχεδιάζονται με τρόπο που να κρύβουν την πολυπλοκότητα και να κάνουν την τεχνολογία εύκολη στη χρήση ακόμη και από μη ειδικούς. Το ιδανικά σχεδιασμένο σύστημα οπτικής επιθεώρησης είναι ρυθμιζόμενο για διαφορετικά προϊόντα και τύπους ελαττωμάτων (ευέλικτα), ικανά να ανιχνεύουν ελαττώματα με ακρίβεια (ευαίσθητα) και ικανά να διακρίνουν σωστά τα ελαττώματα

από τις αποδεκτές αποκλίσεις (επιλεκτικά).

Η προσδοκία για την επίτευξη της υψηλότερης ευαισθησίας και επιλεκτικότητας ταυτόχρονα είναι εκεί που το ιδανικό συναντά την πραγματικότητα.

Οι δύο πιο συνηθισμένοι λόγοι απογοήτευσης στην τεχνολογία επιθεώρησης με μηχανική όραση είναι οι μη ρεαλιστικές προσδοκίες ευαισθησίας, δηλαδή η ικανότητα ανίχνευσης ελαττωμάτων, και η επιλεκτικότητα, δηλαδή την ικανότητα να έχουμε χαμηλό ποσοστό απόρριψης καλών προϊόντων [14].

Η πιο κοινή παρανόηση είναι ότι η εγκατάσταση ενός συστήματος με κάμερα υψηλής ανάλυσης καθιστά την επιθεώρηση εξαιρετικά ευαίσθητη και επιλεκτική. Ωστόσο, η ευαισθησία του συστήματος επιθεώρησης εξαρτάται από πολλές παραμέτρους που δεν είναι πάντα προφανείς. Η οπτική παράμετρος δεν είναι μόνο η ανάλυση, αλλά και ο θόρυβος της κάμερας, η ομοιογένεια του φωτός, η παραμόρφωση του φακού κ.τ.λ. Επιπλέον, δεν υπάρχουν απλώς οπτικές παράμετροι στα συστήματα οπτικής επιθεώρησης. Η σταθερότητα και η επαναληψιμότητα του μηχανικού χειρισμού του προϊόντος και η σωστή ανάλυση εικόνας είναι εξίσου σημαντικές.

Ένας ελαττωματικός ή ατελής σχεδιασμός σε ένα μόνο από τα τμήματα του συστήματος επιθεώρησης μπορεί να οδηγήσει σε υψηλότερη αβεβαιότητα μέτρησης και συνεπώς χαμηλότερα ποσοστά ανίχνευσης και αύξηση του ποσοστού απόρριψης καλών προϊόντων. Επιπλέον, το ποσοστό απόρριψης είναι ένας συμβιβασμός μεταξύ της ευαισθησίας σε σφάλματα, και της επιλεκτικότητας. Επειδή η απόρριψη μη ελαττωματικών προϊόντων σημαίνει άμεση απώλεια εσόδων της επιχείρησης, είναι σημαντικό να μειωθεί όσον τον δυνατόν.

1.3.6 - Το μέλλον για τα συστήματα όρασης

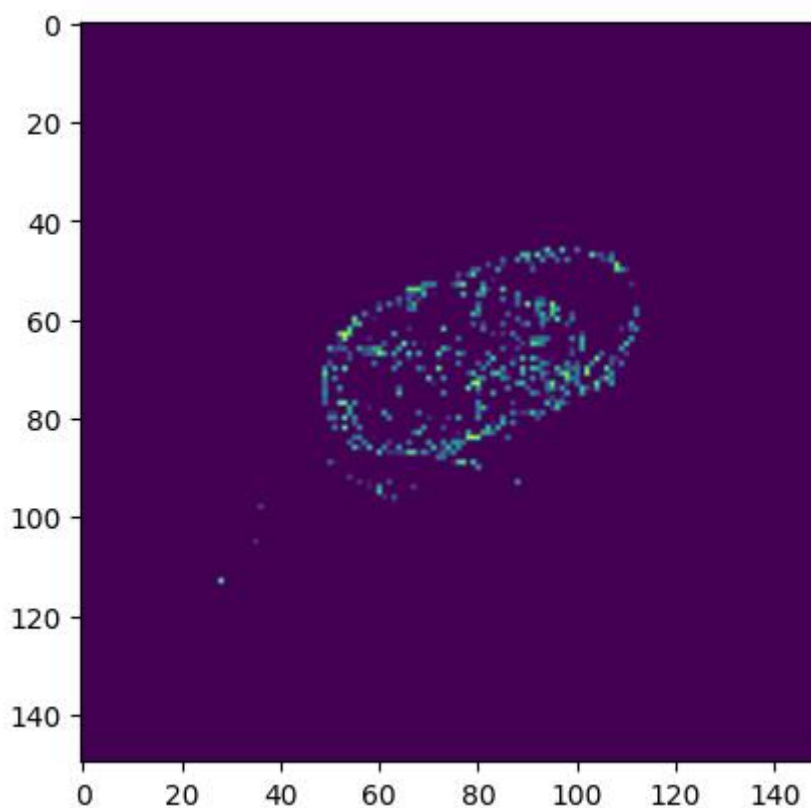
Από την πλευρά του μηχανήματος, οι εξελίξεις εξαρτημάτων προσφέρουν εξαιρετικά βελτιωμένες πρώτες ύλες, όπως μεγαλύτερη ποικιλία καμερών που μπορούν να χρησιμοποιηθούν για τη δημιουργία πολύ συγκεκριμένων λύσεων λήψης εικόνας, νέους φακούς, περίπλοκη ρομποτική κ.τ.λ.

Αν και η μηχανική όραση προϋπήρξε της υπολογιστικής όρασης, είναι σημαντικό

να συγχέονται οι τεχνολογίες. Αφού η υπολογιστική όραση μπορεί να ενισχύσει της ικανότητες της μηχανικής όρασης.

Η επιχειρηματική αξία της μηχανικής όρασης δεν μπορεί να αγνοηθεί αφού το Bloomberg προτείνει ότι η παγκόσμια αγορά μηχανικής όρασης θα μπορούσε να αυξηθεί και να φτάσει τα 18 δισεκατομμύρια δολάρια έως το 2025, η Allied Market Research αναμένει ότι η παγκόσμια αγορά συστημάτων μηχανικής όρασης θα αυξηθεί κοντά στα 74.9 δισεκατομμύρια δολάρια έως το 2027 [10].

1.3.7 - Συμπέρασμα



Εικόνα 1.10. Η εικόνα 1.7 περασμένο με τον canny edge detector [15] μπορούμε να δούμε ότι έχει μολυνθεί κάτι που δύσκολα παρατηρείται με το ανθρώπινο μάτι στην αρχική εικόνα.

Τα συστήματα υπολογιστικής όρασης μπορούν να βελτιώσουν τις διαδικασίες ποιοτικού ελέγχου, να ανιχνεύσουν γρήγορα σφάλματα, να μειώσουν τις ανακλήσεις προϊόντων και να βελτιώσουν την ασφάλεια.

Η παραγωγή βασίζεται στην προηγμένη μηχανική όραση για την παρασκευή των προϊόντων τους ώστε να ανταποκριθούν στις αυξανόμενες απαιτήσεις όγκου για

επιθεώρηση προϊόντων της. Τα συστήματα όρασης θα χρησιμοποιηθούν εκτενέστερα στη παραγωγική βιομηχανία για την βελτιστοποίηση των διαδικασιών παραγωγής. Η αυξημένη ακρίβεια και αποτελεσματικότητα στα συστήματα υπολογιστικής όρασης θα ωφελήσει αναμφίβολα τους παρασκευαστές και τους πελάτες τους.

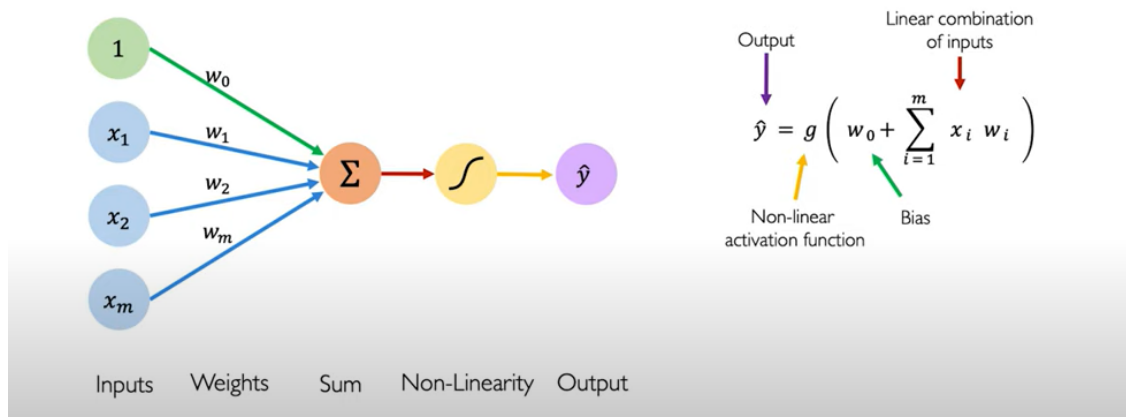
Κεφαλαίο 2 - Βαθιά Νευρωνικά Δίκτυα

2.1 - Περίληψη

Από την δημιουργία τους τα νευρωνικά δίκτυα έχουν εγκατασταθεί ως το κύριο εργαλείο για την προσομοίωση της ανθρώπινης νοημοσύνης στους υπολογιστές. Με την βοήθεια παραδοσιακών μεθόδων αυτοματισμού και της μηχανικής όρασης έχουν την ικανότητα να αναβαθμίσουν όλους τους βιομηχανικούς και παραγωγικούς κλάδους. Συγκεκριμένα θα αναλυθούν τα κύρια τμήματα της υπολογιστικής όρασης η οποία έχει πολλή καλά μελετημένες και αποδεδειγμένες εφαρμογές πάνω από 1 δεκαετία.

2.2 - Το perceptron

Η βαθιά μάθηση ως ένα υποσύνολο της τεχνητής νοημοσύνης έχει σκοπό της εκπαίδευση των υπολογιστών προσομοιάζοντας την ανθρώπινη νοημοσύνη. Το πιο θεμελιώδες στοιχείο της βαθιάς μάθησης είναι τα νευρωνικά δίκτυα αποτελούμενα από πολλούς νευρώνες τα λεγόμενα Perceptron. Ο νευρώνας αποτυπώνει έναν γραμμικό ταξινομητή και αποτελείται από 3 τμήματα την είσοδος (input), το βάρος-πόλωση (weight-bias), την συνάρτηση ενεργοποίησης (activation function). Η είσοδος πολλαπλασιάζεται με ένα βάρος, το γινόμενο προσθέτετε με την πόλωση, το τελικό αποτέλεσμα είναι είσοδος για μια συνάρτηση ενεργοποίησης η έξοδος της οποίας θα διεγύρη τον νευρώνα.



Εικόνα 2.1. Το perceptron [16].

2.3 - Πλήρως Συνδεδεμένα Πυκνά Στρώματα (Fully Connected Dense Layer)

Τα πλήρως συνδεδεμένα πυκνά στρώματα είναι μια γραμμική εξίσωση που ενώνει όλες της εισόδους νευρώνες με όλες τις εξόδους νευρώνες. Ένα πλήρως συνδεδεμένο πυκνό στρώμα, γνωστό και ως γραμμικό στρώμα, μπορεί να περιγραφεί μαθηματικά ως εξής:

Δεδομένου ενός διανύσματος εισόδου x της διάστασης N και ενός διανύσματος εξόδου y της διάστασης M , ένα πυκνό στρώμα μπορεί να αναπαρασταθεί ως πολλαπλασιασμός πίνακα ακολουθούμενο από μια προσθήκη όρων πόλωσης (bias). Ας συμβολίσουμε τον πίνακα βάρους ως W με διαστάσεις $M \times N$ και το διάνυσμα πόλωσης ως b με διαστάσεις M .

Η έξοδος y του πυκνού στρώματος μπορεί να υπολογιστεί ως:

$$y = Wx + b \quad (1)$$

Εδώ:

- W είναι ο πίνακας βάρους, όπου κάθε σειρά αντιστοιχεί στα βάρη που σχετίζονται με έναν συγκεκριμένο νευρώνα στο επίπεδο εξόδου και κάθε στήλη αντιστοιχεί στις συνδέσεις από τους νευρώνες εισόδου.

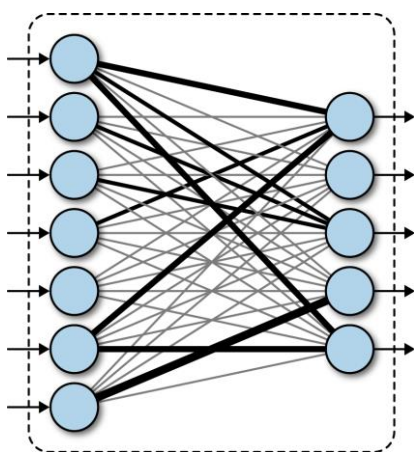
- x είναι το διάνυσμα εισόδου, που περιέχει τις ενεργοποιήσεις από τους νευρώνες

του προηγούμενου στρώματος.

- b είναι το διάνυσμα πόλωσης, που περιέχει όρους πόλωσης για κάθε νευρώνα στο επίπεδο εξόδου.

Ο πολλαπλασιασμός του πίνακα Wx καταγράφει το σταθμισμένο άθροισμα των ενεργοποιήσεων εισόδου, αντιπροσωπεύοντας την επίδραση κάθε νευρώνα εισόδου σε κάθε νευρώνα εξόδου. Η προσθήκη του διανύσματος πόλωσης b εισάγει μια μετατόπιση, επιτρέποντας στο μοντέλο να λαμβάνει υπόψη τις εγγενείς προκαταλήψεις ή παρεμβολές στα δεδομένα.

Μέσω αυτής της μαθηματικής διατύπωσης, ένα πλήρως συνδεδεμένο πυκνό στρώμα μετατρέπει τα δεδομένα εισόδου σε αναπαραστάσεις υψηλότερων διαστάσεων, επιτρέποντας στα νευρωνικά δίκτυα να μαθαίνουν και να μοντελοποιούν πολύπλοκες σχέσεις που υπάρχουν στα δεδομένα.



Εικόνα 2.2. Διαγραμματική αντιπροσωπεύσει του perceptron. Με γραμμες συμβολίζονται τα βάρη ενώ με κύκλους τους νευρώνες (nodes perceptron) [17].

Εάν $y_i \in \mathfrak{R}$ είναι η i -οστη έξοδο από έναν πλήρες συνδεδεμένο επίπεδο, τότε $y_i \in \mathfrak{R}$ υπολογίζεται ακολούθως:

$$y_i = \sigma(W_1X_1 + \dots + W_mX_m) \quad (2)$$

Με στην μη γραμμική συνάρτηση ενεργοποίησης, στην συγκεκριμένη περίπτωση η σιγμοειδής συνάρτηση.

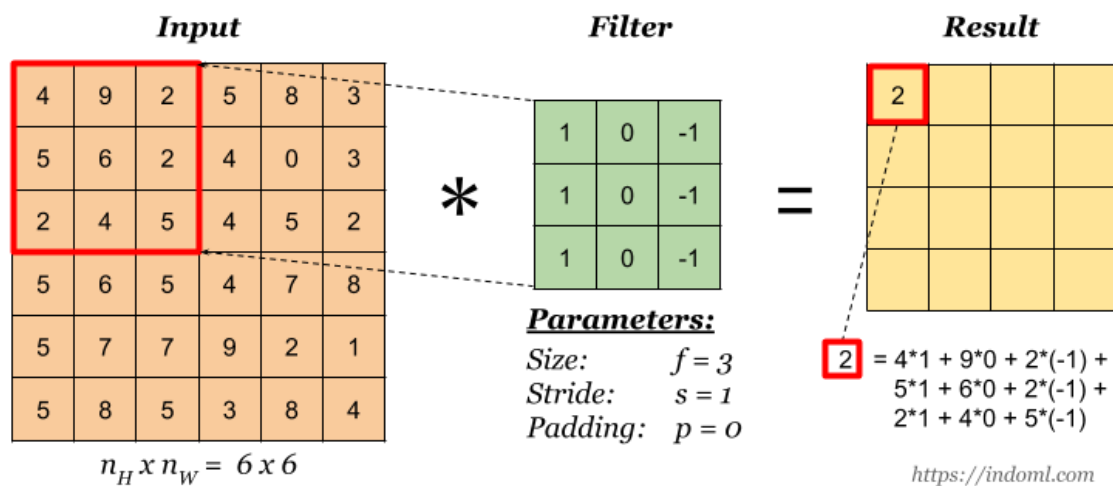
2.2 - Συνελικτικό Νευρωνικό Δίκτυο

Το συνελικτικό νευρωνικό δίκτυο είναι ένας τύπος τεχνητού νευρωνικού συστήματος. Παρατάθηκε αρχικά τη δεκαετία του 1980 από τον LeCun & co [18], ο σκοπός του είναι να “σπάσει” την εικόνα σε πολλά μέρη και να διαχωρίσουν το ένα με το άλλο. Ένα πλεονέκτημα χρήσης των Συνελικτικών Νευρωνικών Δικτύων είναι ότι αξιοποιούν τη χρήση τοπικής χωρικής συνοχής στις εικόνες εισόδου, που επιτρέπουν να έχουν λιγότερα βάρη καθώς ενεργοποιούνται ορισμένες μόνο παραμέτρους. Αυτή η διαδικασία είναι πολύ αποτελεσματική όσον αφορά τη μνήμη και την πολυπλοκότητα. Τα βασικά δομικά στοιχεία του συνελικτικού νευρωνικού δικτύου έχουν ως εξής:

2.2.1 - Επίπεδο συνέλιξης

Σε συνελικτικό επίπεδο, μια μήτρα ο επώνυμος πυρήνας (kernel) περνά πάνω από τον πίνακα εισόδου για να δημιουργήσει ένα χάρτη χαρακτηριστικών για το επόμενο επίπεδο. Εκτελούμε μια μαθηματική πράξη που ονομάζεται συνέλιξη, με ολίσθηση του πίνακα πυρήνα πάνω από τον πίνακα εισόδου. Σε κάθε τοποθεσία, ένας πολλαπλασιασμός πίνακα βάσει στοιχείων εκτελείται και αθροίζει το αποτέλεσμα στον χάρτη χαρακτηριστικών. Η συνέλιξη είναι ένα εξειδικευμένο είδος γραμμικής λειτουργίας που χρησιμοποιείται ευρέως σε διάφορους τομείς, συμπεριλαμβανομένων επεξεργασία σημάτων. Η συνέλιξη μπορεί εφαρμόζεται σε περισσότερους από 1 άξονες δηλαδή αν έχουμε εισόδους δύο διαστάσεων όπως μια εικόνας, I , και ένα δύο διαστάσεων φίλτρο πυρήνα, K , η σύνθετη εικόνα υπολογίζεται ως ακολούθως:

$$S(i, j) = \sum_m \sum_n I(m, n)k(i - m, j - n) \quad (3)$$

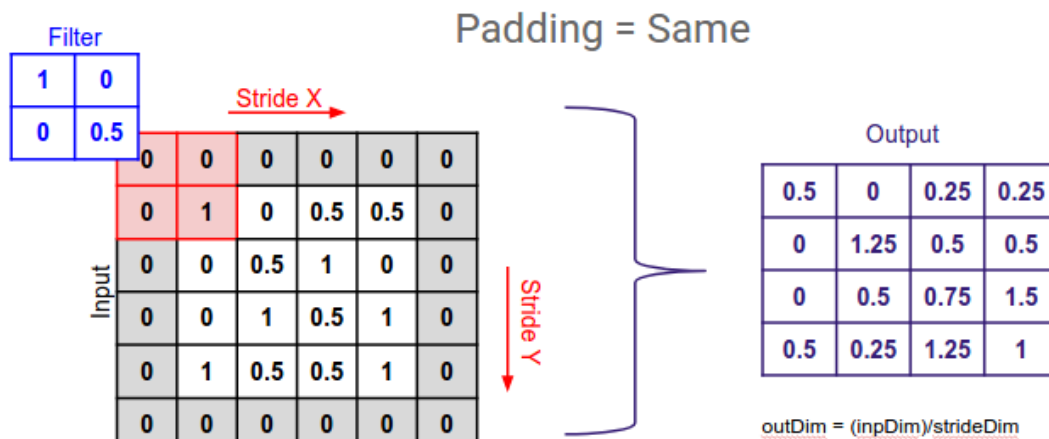


Εικόνα 2.3. Πολλαπλασιασμός ανά στοιχείο και άθροιση των αποτελεσμάτων στον χάρτη χαρακτηριστικών σε συνελκτικό επίπεδο [19].

2.2.2 - Padding

Το padding είναι ένας όρος που σχετίζεται με τα συνελκτικά νευρωνικά δίκτυα καθώς αναφέρεται στην ποσότητα των pixel που προστίθενται σε μια εικόνα όταν υποβάλλεται σε επεξεργασία από τον πυρήνα ενός CNN. Για παράδειγμα, εάν η συμπλήρωση σε ένα CNN είναι μηδενική, τότε κάθε τιμή pixel που προστίθεται θα έχει τιμή μηδέν.

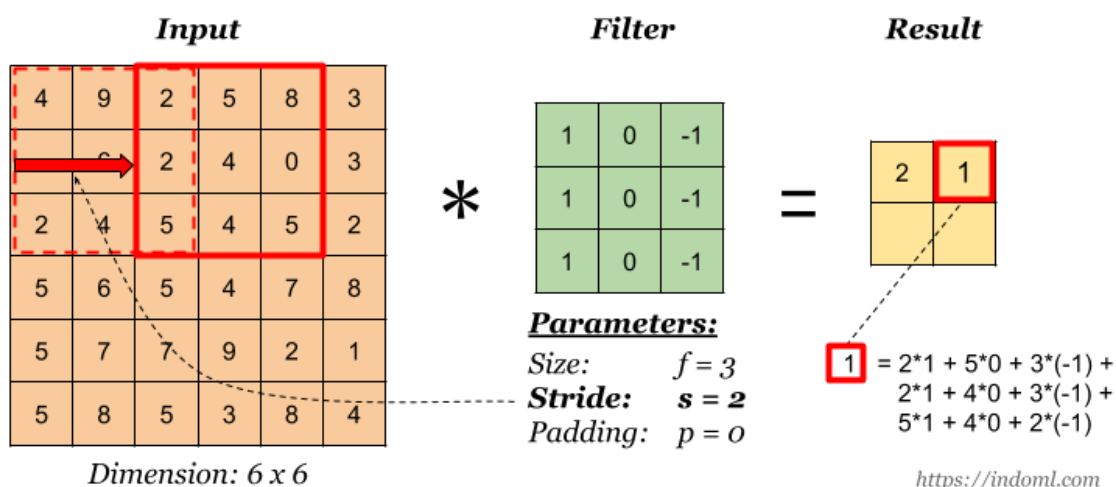
Το padding λειτουργεί επεκτείνοντας την περιοχή της οποίας ένα συνελκτικό νευρωνικό δίκτυο επεξεργάζεται μια εικόνα. Ο πυρήνας είναι το φίλτρο των συνελκτικών νευρωνικών δικτύων που κινείται στην εικόνα, σαρώνοντας κάθε pixel και μετατρέποντας τα δεδομένα σε μικρότερη, ή μερικές φορές μεγαλύτερη, μορφή. Προκειμένου να βοηθηθεί ο πυρήνας στην επεξεργασία της εικόνας, προστίθεται padding στο πλαίσιο της εικόνας για να δοθεί περισσότερος χώρος στον πυρήνα για να καλύψει την εικόνα. Η προσθήκη padding σε μια εικόνα που επεξεργάζεται ένα CNN επιτρέπει την ακριβέστερη ανάλυση των εικόνων.



Εικόνα 2.4. Αποτέλεσμα του padding "same" η διάσταση του αρχικού πίνακα δεν μεταβάλλεται [19].

2.2.3 - Διασκελισμός (Stride)

Στην συνέλιξη με βήμα strides convolution η μάσκα δεν εφαρμόζεται σε όλες τις δυνατές θέσεις της εικόνας αλλά σε θέσεις που απέχουν μεταξύ τους απόσταση ίση με το βήμα που έχει επιλεγθεί. δηλαδή η συνέλιξη με βήμα έχει σκοπό τη μείωση των πράξεων επειδή η μάσκα εφαρμόζεται λιγότερες φορές σε σύγκριση με την πλήρη συνέλιξη. Το κόστος αυτής της μείωσης των πράξεων είναι ότι ο χάρτης χαρακτηριστικών προκύπτει έχει μικρότερη ανάλυση. Ωστόσο πολλές φορές αυτό δεν είναι σημαντικό πρόβλημα και προτιμάται η χρήση βημάτων μεγαλύτερο από ένα επειδή έτσι επιτυγχάνεται σημαντική μείωση των υπολογισμών χωρίς ιδιαίτερα μεγάλη μείωση της επίδοσης.



Εικόνα 2.5. Το βήμα με το οποίο "κινείται" το φίλτρο πάνω στον πίνακα χαρακτηριστικών [19].

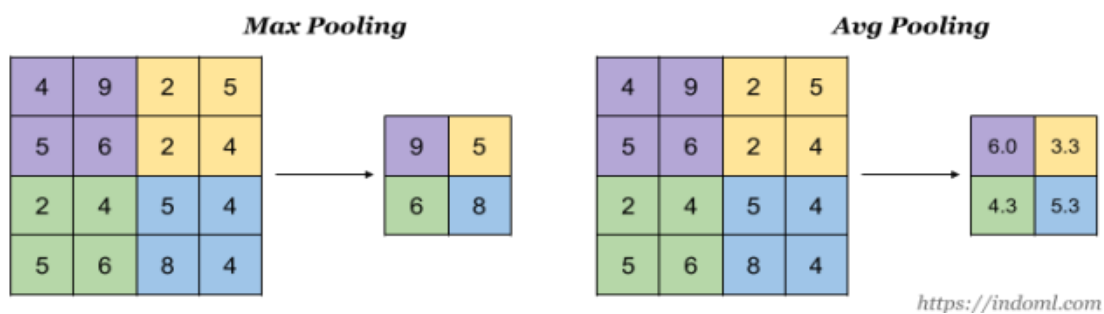
2.2.4 - Στρώμα υποδειματοληψίας (pooling layer)

Σκοπός του στρώματος αυτού είναι να κάνει το σύστημα λιγότερο ευαίσθητο σε μικρές μετατοπίσεις των αντικειμένων της εικόνας, καθώς επίσης να αφαιρέσει τις λεπτομέρειες, χωρίς ωστόσο να προκληθεί απώλεια της ικανότητας διαχωρισμού των αντικειμένων. Προσφέρεται έτσι μια συμπίεση των δεδομένων που οδηγεί αναπόφευκτα σε μια μείωση των πράξεων.

Στο στρώμα υποδειματοληψίας η έξοδος κάθε νευρώνα είναι η σύνοψη των εξόδων των νευρώνων από μία συγκεκριμένη γειτονιά του προηγούμενου στρώματος δηλαδή από το λεγόμενο υποδεκτικό πεδίο του προηγούμενου στρώματος. Με τον όρο σύνοψη εννοούμε μία συνάρτηση που παίρνει τις τιμές των νευρώνων του αποδεικτικού πεδίου και επιστρέφει μία συνοπτική τιμή. Μία από τις πιο απλές συναρτήσεις είναι η μέση τιμή, στην περίπτωση αυτή λέμε ότι έχουμε υποδειματοληψία μέσης τιμής average pooling. Μία άλλη δημοφιλής επιλογή είναι υποδειματοληψία μέγιστης τιμής (max pooling), όπου αντί για τη μέση τιμή παίρνουμε τη μέγιστη τιμή των νευρώνων του υποδεικτικού πεδίου. Η υποδειματοληψία μέγιστης τιμής είναι μη γραμμική και συνήθως προσφέρει καλύτερη αναπαράσταση των χαρακτηριστικών σε μειωμένη διάσταση. Για τον λόγο αυτό προτιμάται συχνά σε σχέση με την υποδειματοληψία μέσης τιμής. Ας σημειωθεί ότι σε αντίθεση με το τι συμβαίνει στα δίκτυα MLP στα στρώματα συνελιξής με υποδειματοληψία οι νευρώνες δεν συνδυάζονται με όλους τους νευρώνες του προηγούμενου στρώματος. Αυτό έχει δύο βασικές συνέπειες:

α) Μειώνει το πλήθος των παραμέτρων που απαιτούνται να εκπαιδευτούν. Επειδή ο χώρος των παραμέτρων είναι μικρότερος, ο χώρος αναζήτησης βέλτιστων τιμών είναι μικρότερος και εκπαίδευση γίνεται ταχύτερα.

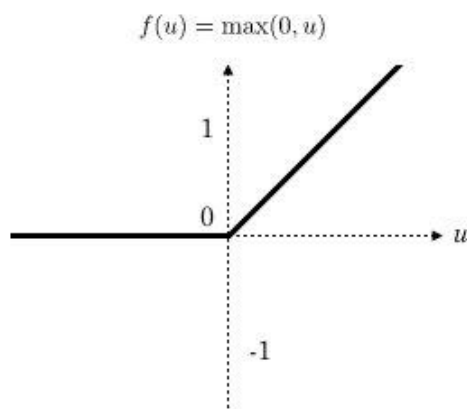
β) Βοηθάει την επίδραση του μοντέλου διότι επιτρέπει την αναγνώριση χαρακτηριστικών γωνίες, ακμές κ.λ.π. οπουδήποτε και αν βρίσκονται στην εικόνα.



Εικόνα 2.6. α) Υποδειγματοληψία με μέγιστης τιμής. β) Υποδειγματοληψία με μέση τιμή [19].

2.2.5 - Μη γραμμική συνάρτηση ενεργοποίησης νευρώνα (ReLU)

Συνάρτηση ράμπας γνωστή και ως ανορθωμένη γραμμική μονάδα είναι μια μη γραμμική συνάρτηση που εξάγει την είσοδο εάν είναι θετική, διαφορετικά θα την μηδενισει.



Εικόνα 2.7. ReLU συνάρτηση ενεργοποίησης.

Κεφάλαιο 3 – Εφαρμογή της βαθιας μαθησης στην κατηγοριοποίηση ελλωματικών χαπιών στην φαρμακοβιομηχανία

3.1 – Περίληψη εφαρμογής

Στόχος της παρούσας εφαρμογής είναι η ανάπτυξη ενός μοντέλου βαθιάς μάθησης με υπάρχουσα εργαλεία και δεδομένα ώστε να επιτευχθεί ακριβή κατηγοριοποίηση ελαττωματικών χαπιών, με οικονομικό τρόπο, κατά τον ποιοτικό έλεγχο μιας γραμμής παραγωγής. Η εφαρμογή χρησιμοποιεί μεθόδους βαθιών συνελκτικών δικτύων (Deep Convolutional Neural Networks) και θα γίνει σύγκριση δημοφιλών μοντέλων κατηγοριοποίησης εικόνων. Για την εκπαίδευση του μοντέλου χρησιμοποιείται η νεφοϋπολογιστική πλατφόρμα της Google Colab όπου διατίθενται υπολογιστική πόροι. Η εφαρμογή θα πραγματοποιηθεί στην γλώσσα προγραμματισμού python με τις open-source βιβλιοθήκες , tensorflow, numpy, scikit-learn, matplotlib, seaborn, pillow.

3.2 – Λόγος Εφαρμογής

Οι ταμπλέτες είναι μια από τις πιο ευρεία διαδεδομένες μορφές φαρμακευτικών αγωγών από το στόμα. Οι ταμπλετες μπορούν να έχουν πολλές ατέλειες και ελαττώματα που μπορούν να βρεθούν στην επιφάνεια ή στις εσωτερικές στρώσεις του προϊόντος κατά την διάρκεια της δημιουργίας της. Επειδή μια και μόνο ατελία στην εμφάνιση ενός μιας ταμπλέτας είναι αρκετό σε μια συσκευασία να εγείρει σοβαρές αμφιβολίες σχετικά με την ακεραιότητα και την ποιότητα ολόκληρου του προϊόντος, οι φαρμακευτικές εταιρείες προσπαθούν συνεχώς να εξασφαλίσουν υψηλή ποιότητα των προϊόντων τους. Ο οπτικός ποιοτικός έλεγχος μπορεί να γίνει με διάφορες στατιστικές τεχνικές ή με οπτική επιθεώρηση και ταξινόμηση, είτε χειροκίνητα είτε αυτοματοποιημένα. Λόγω του γεγονότος ότι τα συστήματα στατιστικής δειγματοληψίας, που εκτιμούν τη συνολική ποιότητα μιας δεδομένης παρτίδας δισκίων σε ένα ορισμένο επίπεδο εμπιστοσύνης, δεν μπορούν να εξασφαλίσουν την απαιτούμενη ποιότητα κάθε ταμπλέτας, αντικαθίστανται από οπτική επιθεώρηση και ταξινόμηση με μεθόδους υπολογιστικής όρασης. Δεδομένου ότι η χειροκίνητη οπτική επιθεώρηση μεγάλων παρτίδων δισκίων είναι υποκειμενική, αναξιόπιστη, αργή, κουραστική και δαπανηρή, τα αυτοματοποιημένα συστήματα οπτικής επιθεώρησης ταμπλετών χρησιμοποιούνται πιο συχνά. Στο Παράρτημα 1 περιγράφονται τα 13 πιο κοινά ελαττώματα μέσω τον οποίον μπορούμε να προβλέψουμε σε πιο σημείο της διαδικασίας (process related causes) και σύνθεσης (formulation related causes) υπάρχει

πρόβλημα [29].

Στην συγκεκριμένη εφαρμογή, οι λόγοι εκτέλεσης του ποιοτικού ελέγχου με νευρωνικά δίκτυα είναι:

α) Μεγάλη ακρίβεια και αξιοπιστία στην εύρεση ελαττωματικών προϊόντων. Προσπαθούμε να βρεθούν όσον τον δυνατόν πιο ακριβείς τεχνικές επιθεώρησης ανώτερες του ανθρώπινου ειδικευμένου αξιολογητή, ο οποίος επηρεάζεται από τις συνθήκες αξιολόγησης (φωτεινότητα, θερμοκρασία) και την σωματική κούραση που μειώνει την συγκέντρωση και άρα τα αποτελέσματα.

β) Θέλουμε το σύστημα όχι μόνο να ανιχνεύει σφάλματα με αυτόματο τρόπο αλλά ταυτόχρονα να έχει αυτονομία δηλαδή να μπορεί να αξιολογεί τα αποτελέσματα και να δρα αναλόγως χωρίς ανθρώπινο παρατηρητή που να χρειάζεται να παρέμβει.

3.2 – Τα δεδομένα

Τα δεδομένα που χρησιμοποιήθηκαν για την εκπαίδευση και πρόβλεψη του μοντέλου ανίχνευσης ανωμαλιών στα χάπια ανήκει στην MVTec και είναι μέρος από μιας μεγάλης βιβλιοθήκης με 5000 υψηλής ανάλυσης εικόνες εστιασμένες σε διάφορους βιομηχανικούς κλάδους [21]. Επιλέχθηκε έναν μέρος του συνόλου των φωτογραφιών όπου περιλάμβανε 434 δια χειρός σχολιασμένες εικόνες (hand-annotated) οι οποίες είχαν ορθές εικόνες με την ένδειξη (labeled) Good και προβληματικές εικόνες με την ένδειξη Anomaly.

3.3 – Τα μοντέλα

Το MobileNetV2 είναι πολύ παρόμοιο με το αρχικό MobileNet, εκτός από το ότι χρησιμοποιεί ανεστραμμένα υπολειμματικά μπλοκ με χαρακτηριστικά συμφόρησης. Έχει δραστικά χαμηλότερο αριθμό παραμέτρων από το αρχικό MobileNet. Τα MobileNets υποστηρίζουν οποιοδήποτε μέγεθος εισόδου μεγαλύτερο από 32 x 32, με μεγαλύτερα μεγέθη εικόνας που προσφέρουν καλύτερη απόδοση. [50]

Τα μοντέλα VGG16, VGG19 που χρησιμοποιήθηκαν, Πολύ Βαθύ Συνελικτικό Νευρωνικό Δίκτυο (Very Deep Convolutional Networks for Large-Scale Image Recognition), περιγράφηκαν από τους Karen Simonyan, Andrew Zisserman το 2014 [22].

Το μοντέλο ResNet50 που χρησιμοποιήθηκε, Deep Residual Learning for Image

Recognition, περιγράφηκε από τους Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun το 2015 [38].

Το μοντέλο Inception V3 που χρησιμοποιήθηκε, Rethinking the Inception Architecture for Computer Vision, περιγράφηκε από τους Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna το 2015 [39].

Το μοντέλο EfficientNetB0 που χρησιμοποιήθηκε, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, περιγράφηκε από τους Mingxing Tan, Quoc V. Le το 2019 [40].

Τα μοντέλα είναι προεκπαιδευμένα (pretrained) με βάση του ερευνητικής βάσης δεδομένων ImageNet [23] , έγινε τεχνική της επανεκπαίδευσης, διότι έχει εύρεθει ότι καθώς το μοντέλο εκπαιδεύεται σε ένα τεράστιο σύνολο δεδομένων, έχει μάθει μια καλή αναπαράσταση χαρακτηριστικών χαμηλού επιπέδου όπως το χωρικό, άκρες, περιστροφή, φωτισμός, σχήματα και αυτά τα χαρακτηριστικά μπορεί να τα μοιράζεται μεταξύ άλλων δεδομένων και άρα να καταστεί δυνατή η μεταφορά γνώσης που έχει την δυνατότητα να λειτουργήσει ως εξαγωγή χαρακτηριστικών για νέες εικόνες σε διαφορετικά προβλήματα υπολογιστικής όραση.

Η εξαγωγή των συμπερασμάτων για κάθε μοντέλο έγινε με ένα την χρήση ενός νευρώνα Dense(1) αφού έχουμε μόνο δύο κατηγορίες good, anomaly και δεν είναι απαραίτητη χρήση συνάρτησης ενεργοποίησης αφού η τιμή προβλέψεις θα προβλέψει κατηγορία 1 εάν ο αριθμός είναι θετικός και κατηγορία 0 εαν ο αριθμός είναι αρνητικός [41].

3.4 - Διαδικασία κατασκευής

3.4.1 - Προετοιμασία Δεδομένων

Αρχικά κατεβάζουμε τα αρχεία με τις εικόνες με την βοήθεια του wget το οποίο είναι ένα ελεύθερο λογισμικό για την ανάκτηση αρχείου μέσω των πιο γνωστών πρωτοκόλλων όπως HTTP, HTTPS, FTP, FTPS κ.τ.λ.π. Με το command tar - xf αποσυμπιέζουμε το αρχείο.

```
!wget https://www.mydrive.ch/shares/43421/11a215a5749fcfb75e331ddd5f8e43ee/download/420938129-1629953099/pill.tar.xz

--2023-09-22 14:43:18-- https://www.mydrive.ch/shares/43421/11a215a5749fcfb75e331ddd5f8e43ee/download/420938129-1629953099/pill.tar.xz
Resolving www.mydrive.ch (www.mydrive.ch)... 91.214.169.64
Connecting to www.mydrive.ch (www.mydrive.ch)|91.214.169.64|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://stor-ch.mydrive.ch/download/63c8552d9c734450bd5d59bee86acfd96836cd9fc26045dcbd60341c9a4bb268?1629953099 [following]
--2023-09-22 14:43:20-- https://stor-ch.mydrive.ch/download/63c8552d9c734450bd5d59bee86acfd96836cd9fc26045dcbd60341c9a4bb268?1629953099
Resolving stor-ch.mydrive.ch (stor-ch.mydrive.ch)... 91.214.169.64
Connecting to stor-ch.mydrive.ch (stor-ch.mydrive.ch)|91.214.169.64|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 275267152 (263M) [application/x-xz]
Saving to: 'pill.tar.xz'

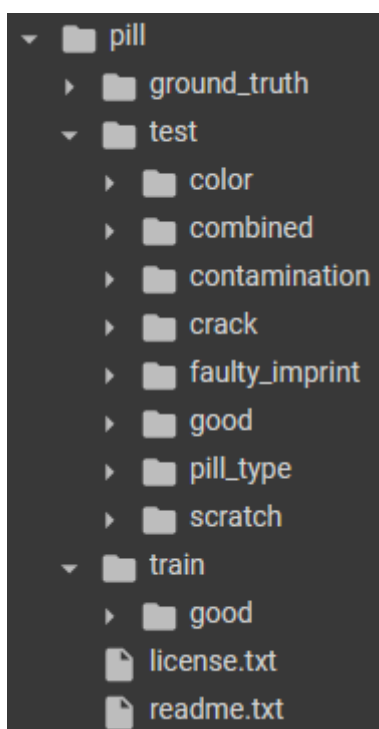
pill.tar.xz      100%[=====>] 262.51M  14.6MB/s  in 18s

2023-09-22 14:43:39 (14.5 MB/s) - 'pill.tar.xz' saved [275267152/275267152]

!tar -xvf pill.tar.xz
```

Εικόνα 3.1. Κώδικας για κατέβασμα και εξαγωγή των δεδομένων.

Παρατηρούμε ότι το path των αρχείων των εικόνων δεν είναι στην επιθυμητό μέρος.



Εικόνα 3.2. Διαδρομή των αρχείων.

Πρέπει να κάνουμε επεξεργασία των path των εικόνων και να τις διαχειριστούμε σε δεδομένα ελέγχου και δεδομένα εκπαίδευσης. Πρώτα μετονομαζουμε τα αρχεία ώστε να μην έχουν το ίδιο όνομα.

```

import os
import random
import shutil

# Specify the folder path where the files are located
# Source directory containing the files you want to split
source_directory1 = '/content/pill/test/colon'
source_directory2 = '/content/pill/test/combined'
source_directory3 = '/content/pill/test/crack'
source_directory4 = '/content/pill/test/faulty_imprint'
source_directory5 = '/content/pill/test/scratch'
source_directory6 = '/content/pill/test/contamination'

source_dir_list = [source_directory1, source_directory2, source_directory3, source_directory4, source_directory5, source_directory6]
for source_directory in source_dir_list:

    # Get a list of all files in the folder
    file_list = os.listdir(source_directory)

    # Define a new name format (e.g., prefix + index + extension)
    new_name_prefix = source_directory.split("/")[-1]
    index = 0 # Start index
    extension = '.png' # Change this to the desired file extension

    # Iterate through the files and rename them
    for old_name in file_list:
        # Construct the new name for the file
        new_name = f"{new_name_prefix}_{index:03d}{extension}" # 04d ensures 4-digit index

        # Create the full file paths for old and new names
        old_file_path = os.path.join(source_directory, old_name)
        new_file_path = os.path.join(source_directory, new_name)

        # Rename the file
        os.rename(old_file_path, new_file_path)

        # Increment the index for the next file
        index += 1

print("Files renamed successfully.")
Files renamed successfully.

```

Εικόνα 3.3. Κώδικας για την μετονομασία των εικόνων.

Μετέπειτα δημιουργούνται 2 φάκελοι που λείπουν ο test/anomaly και ο train/anomaly έπειτα, επιλέγουμε το split ratio ώστε να πάρουμε το επιθυμητό ποσοστό από το καθένα από τις anomaly κατηγορίες και να το τοποθετήσουμε στα κατάλληλους φακέλους για εκπαίδευση και έλεγχο . Όταν πρόκειται για ένα μη ισορροπημένο σύνολο δεδομένων, ο διαχωρισμός εκπαίδευσης/δοκιμής (train/test split) θα πρέπει να εκτελείται με στρωματοποιημένο τρόπο (stratified manner), επομένως τα τμήματα εκπαίδευσης και δοκιμής θα περιέχουν το ίδιο μερίδιο και των δύο κατηγοριών — Good/Anomaly, είναι μία διαδικασία διαμερισμού των προτύπων η οποία φροντίζει το σύνολο εκπαίδευσης και το σύνολο ελέγχου θα περιέχουν δείγματα αναλογικά από όλες τις κλάσεις. Στη συγκεκριμένη περίπτωση επειδή έχουμε μόνο δύο κλάσεις θα φροντίσουμε ώστε ο λόγος των δύο κλάσεων στα δεδομένα εκπαίδευσης να είναι ίσος με το λόγο στα δεδομένα ελέγχου και θα είναι τελικά ίσο με το λόγο των δεδομένων των δύο κλάσεων, συγκεκριμένα 90% Good 10% Anomaly άρα στο σύνολο εκπαίδευσης θα περιέχει 90% Good και 10% Anomaly δεδομένα, παρομοίως θα

μοιράσουμε τα δεδομένα για κάθε διαφορετικό τύπο σφάλματος των ελαττωματικών χαπιών (σπασμένα, βρόμικά, κ.λ.π.) μεταξύ σύνολο εκπαίδευσης και σύνολο ελέγχου. Έτσι ώστε το μοντέλο να εκπαιδευτεί, αλλά και να ελεγχθεί με όλα τα διαφορετικά σφάλματα που μπορεί να προκύψουν.

```
# Destination directories for the two splits
destination_directory1 = '/content/pill/test/anomaly'
destination_directory2 = '/content/pill/train/anomaly'

os.makedirs(destination_directory1, exist_ok=True)
os.makedirs(destination_directory2, exist_ok=True)

# Probability of a file being placed in destination_directory1
split_ratio = 0.1 # You can adjust this as needed

for source_directory in source_dir_list:
    # Get a list of all files in the source directory
    file_list = os.listdir(source_directory)

    # Shuffle the list to randomize the file order
    random.shuffle(file_list)

    # Determine the number of files to put in each destination directory
    split_point = int(len(file_list) * split_ratio)

    # Copy files to the destination directories based on the split ratio
    for i, file_name in enumerate(file_list):
        source_file_path = os.path.join(source_directory, file_name)

        if i < split_point:
            destination_file_path = os.path.join(destination_directory1, file_name)
        else:
            destination_file_path = os.path.join(destination_directory2, file_name)

        # Copy the file to the appropriate destination
        shutil.copy(source_file_path, destination_file_path)

    print(f"Split {len(file_list)} files from {source_directory} into {split_point} files in {destination_directory1} and {len(file_list) - split_point} files in {destination_directory2}.")

Split 25 files from /content/pill/test/color into 2 files in /content/pill/test/anomaly and 23 files in /content/pill/train/anomaly.
Split 17 files from /content/pill/test/combined into 1 files in /content/pill/test/anomaly and 16 files in /content/pill/train/anomaly.
Split 26 files from /content/pill/test/crack into 2 files in /content/pill/test/anomaly and 24 files in /content/pill/train/anomaly.
Split 19 files from /content/pill/test/faulty_imprint into 1 files in /content/pill/test/anomaly and 18 files in /content/pill/train/anomaly.
Split 24 files from /content/pill/test/scratch into 2 files in /content/pill/test/anomaly and 22 files in /content/pill/train/anomaly.
Split 21 files from /content/pill/test/contamination into 2 files in /content/pill/test/anomaly and 19 files in /content/pill/train/anomaly.
```

Εικόνα 3.4. Κώδικας για την δημιουργία των φακέλων και των διαχωρισμό των δεδομένων σε αυτούς.

Τελος διαγράφουμε όλους τους φακέλους πού δεν θα μας είναι απαραίτητη ώστε να έχουμε το directory format προετοιμασίας των δεδομένων για μοντέλα tensorflow μέσω του `tf.keras.utils.image_dataset_from_directory()`.

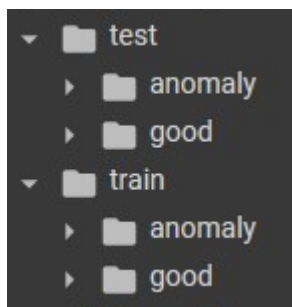
```
folder_to_delete = source_dir_list.append('/content/pill/test/pill_type')
for source_directory in source_dir_list:

    # Specify the path to the folder you want to delete
    folder_to_delete = source_directory

    # Use shutil.rmtree() to delete the folder and its contents
    try:
        shutil.rmtree(folder_to_delete)
        print(f"Folder '{folder_to_delete}' and its contents have been successfully deleted.")
    except Exception as e:
        print(f"Error deleting folder: {e}")

Folder '/content/pill/test/color' and its contents have been successfully deleted.
Folder '/content/pill/test/combined' and its contents have been successfully deleted.
Folder '/content/pill/test/crack' and its contents have been successfully deleted.
Folder '/content/pill/test/faulty_imprint' and its contents have been successfully deleted.
Folder '/content/pill/test/scratch' and its contents have been successfully deleted.
Folder '/content/pill/test/contamination' and its contents have been successfully deleted.
Folder '/content/pill/test/pill_type' and its contents have been successfully deleted.
```

Εικόνα 3.5. Κώδικας για την διαγραφή των φακέλων.



Εικόνα 3.6. Τελικό directory path.

Εισάγουμε όλες τις επιθυμητες βιβλιοθήκες τις οποίες θα χρησιμοποιηθούν για εκπαίδευση και έλεγχο των δεδομένων. Ορίζουμε την διαδρομή των αρχείων εκπαίδευσης και ελέγχου το μεγεθον τον εικόνων και το batch size με το οποίο θα ομαδοποιηθούν τα δεδομένα.

```
import matplotlib.pyplot as plt
import numpy as np
import tensorflow as tf
from tensorflow.keras.applications import VGG16, ResNet50, InceptionV3, EfficientNetB0, VGG19
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.optimizers import Adam

train_dir = '/content/pill/train'
test_dir = '/content/pill/test'

BATCH_SIZE = 10
IMG_SIZE = (224, 224)

train_dataset = tf.keras.utils.image_dataset_from_directory(train_dir,
                                                            shuffle=True,
                                                            batch_size=BATCH_SIZE,
                                                            image_size=IMG_SIZE)

Found 389 files belonging to 2 classes.

test_dataset = tf.keras.utils.image_dataset_from_directory(test_dir,
                                                           shuffle=True,
                                                           batch_size=BATCH_SIZE,
                                                           image_size=IMG_SIZE)

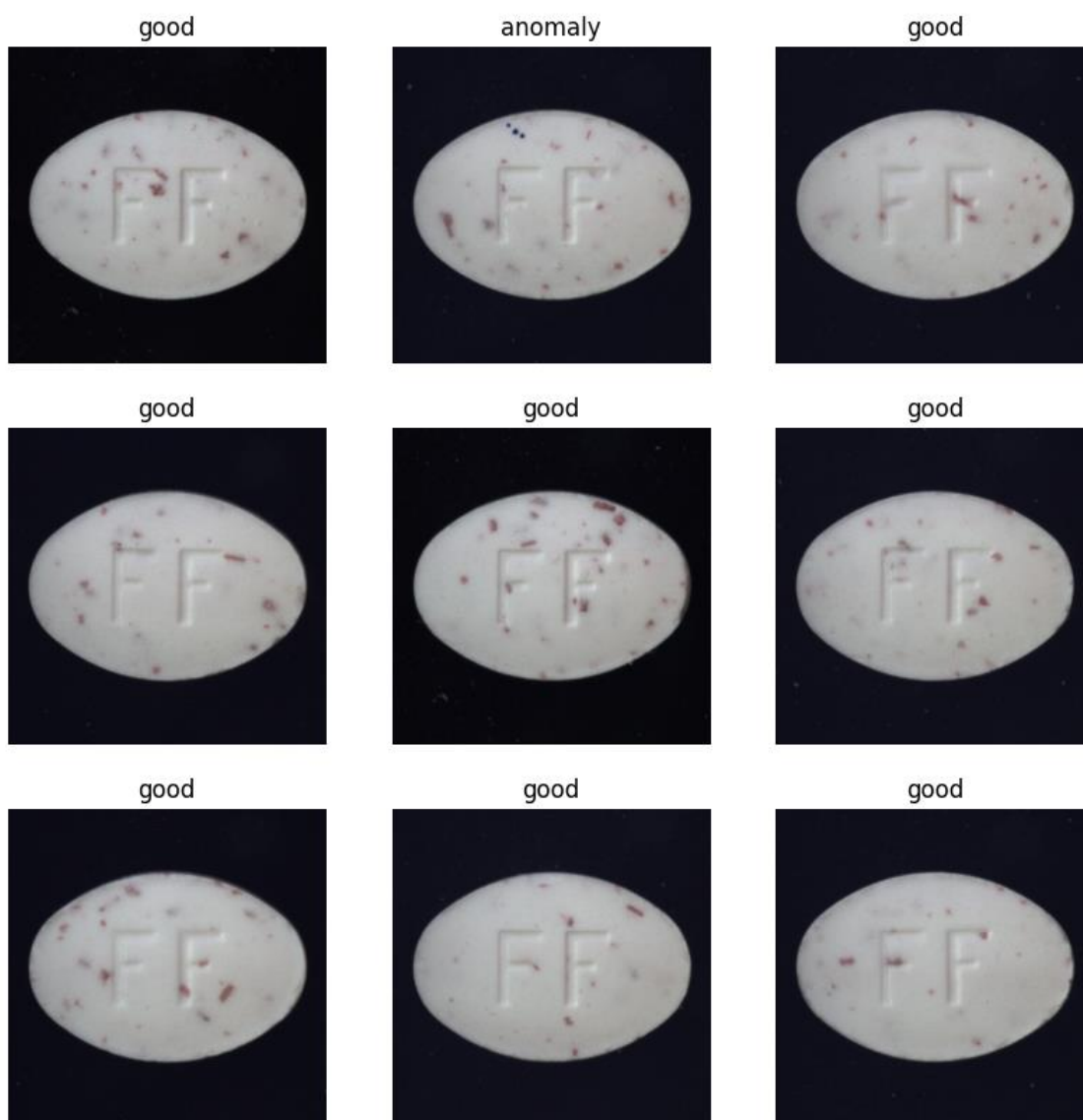
Found 36 files belonging to 2 classes.
```

Εικόνα 3.7. Κώδικας για την εισαγωγή βιβλιοθηκών και χρήση του image_dataset_from_directory.

```
class_names = train_dataset.class_names

plt.figure(figsize=(10, 10))
for images, labels in train_dataset.take(1):
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(class_names[labels[i]])
        plt.axis("off")
```

Εικόνα 3.8. Κώδικας για την γραφικοποίηση των πρώτων 9 εικόνων από τα δεδομένα ελέγχου μαζί με τα label τους.



Εικόνα 3.9. Γραφική αναπαράσταση των 9 πρώτων δεδομένων εκπαίδευσης.

Κάνουμε χρήση του `.prefetch` ώστε να φορτώσουμε τις εικόνες μας στην μνήμη της κάρτας γραφικών κατά την διάρκεια της προεπεξεργασίας και εκμάθησης του μοντέλου ώστε να αυξήσουμε την ταχύτητα εκπαίδευσης και ελέγχου του μοντέλου. Αυτό μπορεί μην είναι πάντα δυνατό να συμβεί αφού πρέπει το μέγεθος των δεδομένων να είναι μικρότερο από το μέγεθος της μνήμης της κάρτα γραφικών. Χρησιμοποιούμε `tf.data.AUTOTUNE` για να πραγματοποιηθεί αυτο με αυτόματο τρόπο.

```
AUTOTUNE = tf.data.AUTOTUNE

train_dataset = train_dataset.prefetch(buffer_size=AUTOTUNE)
test_dataset = test_dataset.prefetch(buffer_size=AUTOTUNE)
```

Τα δεδομένα θα προεπεξεργαστούν προτού να χρησιμοποιηθούν για εκπαίδευση από το μοντέλο. Αρχικά θα αλλάξουμε το μέγεθος των εικόνων από 1.024 x 1.024 pixel στα 224 x 224 pixel αυτό θα αυξήσει σημαντικά την ταχύτητα εκπαίδευσης χωρίς όμως να μειώσει πολύ την την επίδοση του μοντέλου. Σε επόμενη φάση οι εικόνες θα περάσουν από πολλά βήματα μετασχηματισμού τα οποία έχουν σκοπό την αύξηση των δεδομένων, τη βελτίωση στη γενίκευση του μοντέλου και την καλύτερη του επίδοση σε πραγματικές συνθήκες. Συγκεκριμένα θα κάνουμε τυχαία οριζόντια και κατακόρυφη αναστροφή των εικόνων επίσης θα εφαρμόσουμε μία μικρή ποσότητα περιστροφής ώστε να προσομοιώσουμε πιθανά σφάλματα που μπορούν να συμβούν κατά τη διάρκεια της γραμμής παραγωγής. Δεν επιλέχθηκε να εφαρμοστούν μετασχηματισμοί που θα επηρεάσουν τη φωτεινότητα, τα χρώματα και την αντίθεση της εικόνας αφού αυτά είναι συνθήκες που μπορούν να ελεγχθούν πολύ καλά κατά τη διάρκεια της γραμμής παραγωγής και άρα εκτός εάν η εφαρμογή το υποχρεούται δεν χρειάζεται να εφαρμοστούν. Επίσης πρέπει να προσέχουμε ώστε να μην κάνουμε πολύ ακραίους μετασχηματισμούς αφού αυτό μπορεί να επηρεάσει αρνητικά το μοντέλο δίνοντάς του μη ρεαλιστικές συνθήκες που δεν θα βρεθούμε στην πραγματικότητα.

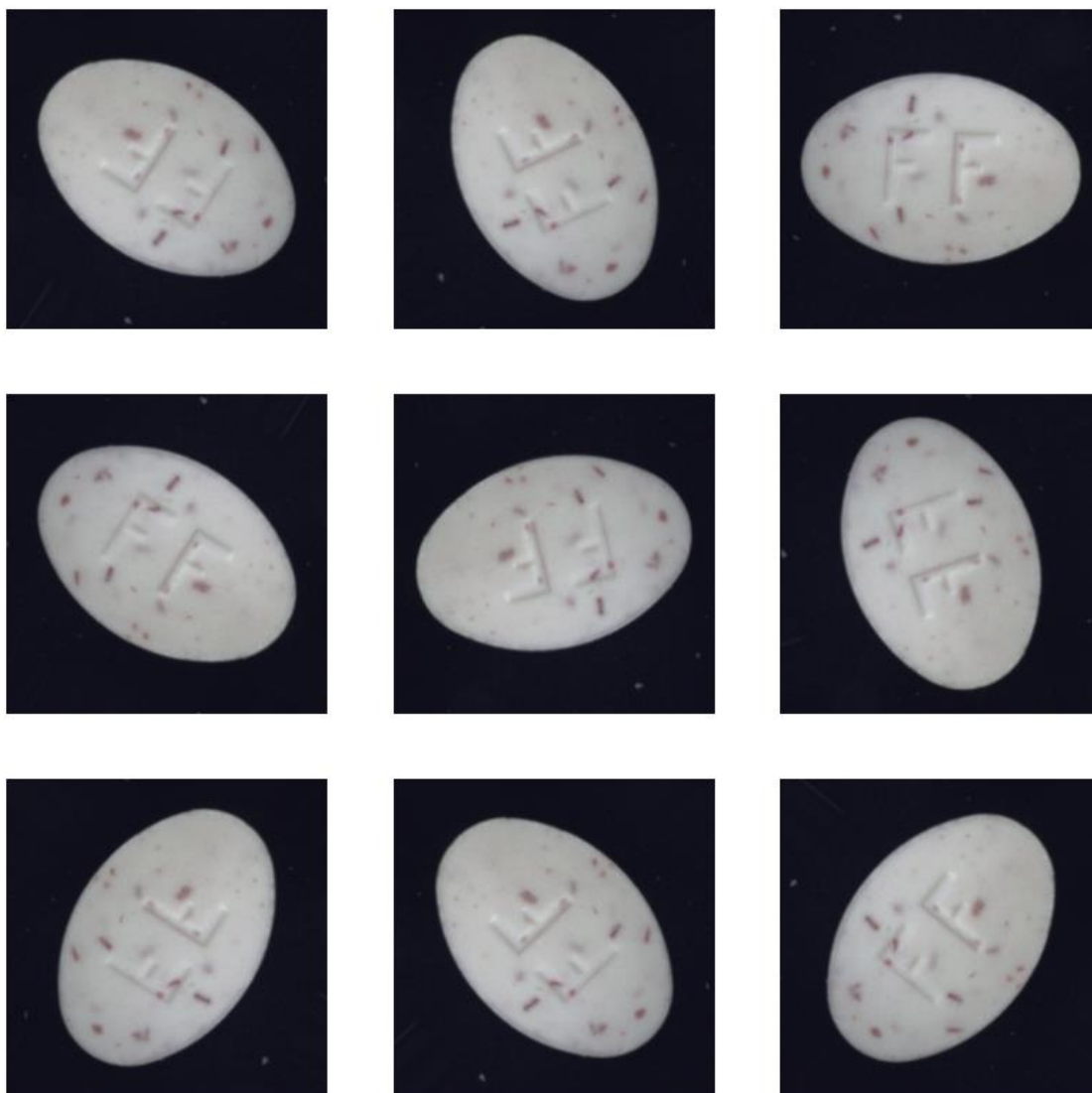
Αλλαζουμε το fill mode σε nearest αντι του default mirror που θα βαλει στην εικόνα τις ακρές του χαπιού

```
[14] data_augmentation = tf.keras.Sequential([
      tf.keras.layers.RandomFlip('horizontal_and_vertical'),
      tf.keras.layers.RandomRotation(0.2, fill_mode='nearest'),
    ])
```

Εικόνα 3.10. Κώδικας για τη προετοιμασία των εικόνων.

```
for image, _ in train_dataset.take(1):
    plt.figure(figsize=(10, 10))
    first_image = image[0]
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        augmented_image = data_augmentation(tf.expand_dims(first_image, 0))
        plt.imshow(augmented_image[0] / 255)
        plt.axis('off')
```

Εικόνα 3.11. Κωδικας για την γραφικοποίηση μιας εικόνας περασμένη πολλές φορές από το data_augmentation στρώμα.



Εικόνα 3.12. Γραφική αναπαράσταση μιας εικόνας από το augmentation layer.

Το κάθε ένα από τα μοντέλα που θα συγκρίνουμε έχουν διαφορετικό τρόπο για την αντιπροσώπευση των εικόνων που θα είναι είσοδος σε αυτά. Στη συγκεκριμένη περίπτωση το `mobilenetv2` περιμένει εικόνες με τιμές pixel ανάμεσα στο $[-1,1]$, ένας τρόπος είναι να προσθέσουμε ένα `rescale` στρώμα το οποίο θα μετασχηματίζει τις τιμές των pixel από $[0,1]$ σε $[-1,1]$, όμως στη συγκεκριμένη εφαρμογή χρησιμοποιήθηκαν οι έτοιμες συναρτήσεις από την Tensorflow οι οποίες επεξεργάζονται τις εικόνες ανάλογα με τον τύπο του μοντέλου που θα χρησιμοποιηθεί.

```
preprocess_input = tf.keras.applications.mobilenet_v2.preprocess_input
```

Εικόνα 3.13. Κώδικας για την επεξεργασία των εικόνων βάση του format που απαιτείται από το μοντέλο.

3.4.2 - Μοντέλο

Κατεβάζουμε το μοντέλο mobilenet v2 και το μετασχηματίζουμε κατάλληλα δηλαδή το input θα πάρει εικόνες μεγέθους 224 pixel θα αφαιρέσουμε το κεφαλή κατηγοριοποίησης το οποίο έρχεται μαζί με το μοντέλο και μας είναι ουσιαστικά άχρηστος στη δική μας εφαρμογή και θα επιλέξουμε το μοντέλο που έχει τα βάρια τα οποία έχουν προσαρμοστεί βάσει του imagenet dataset.

```
IMG_SHAPE = IMG_SIZE + (3,)
base_model = tf.keras.applications.MobileNetV2(input_shape=IMG_SHAPE,
                                                include_top=False,
                                                weights='imagenet')

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mobilenet\_v2/9406464/9406464 [=====] - 0s 0us/step
```

Παρατηρούμε πώς το μοντέλο χωρίς την κεφαλή κατηγοριοποίησης μετατρέπει τις εικόνες από 224x 224x3 σε 7 x7 x 1280 το οποίο είναι η τα εξαγωγήμενα χαρακτηριστικά των δεδομένων.

```
image_batch, label_batch = next(iter(train_dataset))
feature_batch = base_model(image_batch)
print(feature_batch.shape)

(10, 7, 7, 1280)
```

Στην αρχή δεν θα κάνουμε εκπαίδευση στο βασικό μοντέλο άρα το παγώνουμε, θα κάνουμε μόνο εκπαίδευση στη δική μας φτιαγμένη κεφαλή κατηγοριοποιήσεις. Αυτό συμβαίνει γιατί το μοντέλο είναι εκπαιδευμένο δηλαδή έχει ήδη κάποια βασική γνώση στην εξαγωγή πληροφοριών όπως γωνίες και γιατί το δεδομένα μας είναι πολύ λίγα για να εκπαιδύσουμε καλά ολόκληρο το μοντέλο θα κάναμε δηλαδή υπέρ εκπαίδευση του μοντέλου στα δεδομένα μας χάνοντας ουσιαστικά την γενίκευση του περιέχει το μοντέλο. Ακόμα και αργότερα που θα κάνουμε fine-tuning στο μοντέλο και άρα θα εκπαιδύσουμε ένα τμήμα του βασικού μοντέλου με τα δεδομένα μας δεν θα κάνουμε εκπαίδευση σε μερικά στρώματα, όπως το batch normalization το οποίο αν επηρεαστεί κατά τη διάρκεια της εκπαίδευσης θα καταστρέψει ουσιαστικά ότι εξαγωγή χαρακτηριστικών έχει το μοντέλο και άρα θα τα αφήσουμε παγωμένο.

```
base_model.trainable = False
```

Θα κατασκευάσουμε το τελικό μοντέλο θα χρησιμοποιήσουμε το στρώμα data Augmentation που ορίσαμε προηγουμένως θα περαστεί αυτή στο βασικό μας μοντέλο το οποίο κάναμε training = False ώστε να μην επηρεάσουμε το batch normalization που περιέχει το μοντέλο, θα κάνουμε flatten την έξοδο του μοντέλου ώστε να μπορούμε να τα περάσουμε στο δικό μας στρώμα κατηγοριοποιήσεις το οποίο θα κάνει capture τα χαρακτηριστικά που εξάγονται από το μοντέλο, θα περιέχει ένα πλήρες συνδεδεμένο στρώμα με 1.024 νευρώνες το οποίο θα συνδέεται με ένα άλλο πλήρως συνδεδεμένο στρώμα με 512 νευρώνες και τα δύο θα περάσω να πω ένα στρώμα dropout το οποίο ανάλογα με την τιμή που θα το δοθεί θα απενεργοποιηθεί τυχαία ένα ποσοστό των νευρώνων έτσι ώστε να αποφευχθεί η υπέρ εκπαίδευση του μοντέλου. Τελικώς το στρώμα εξόδου αποτελείται από ένα νευρώνα το οποίο θα μας δίνει τιμές 0 ή 1.

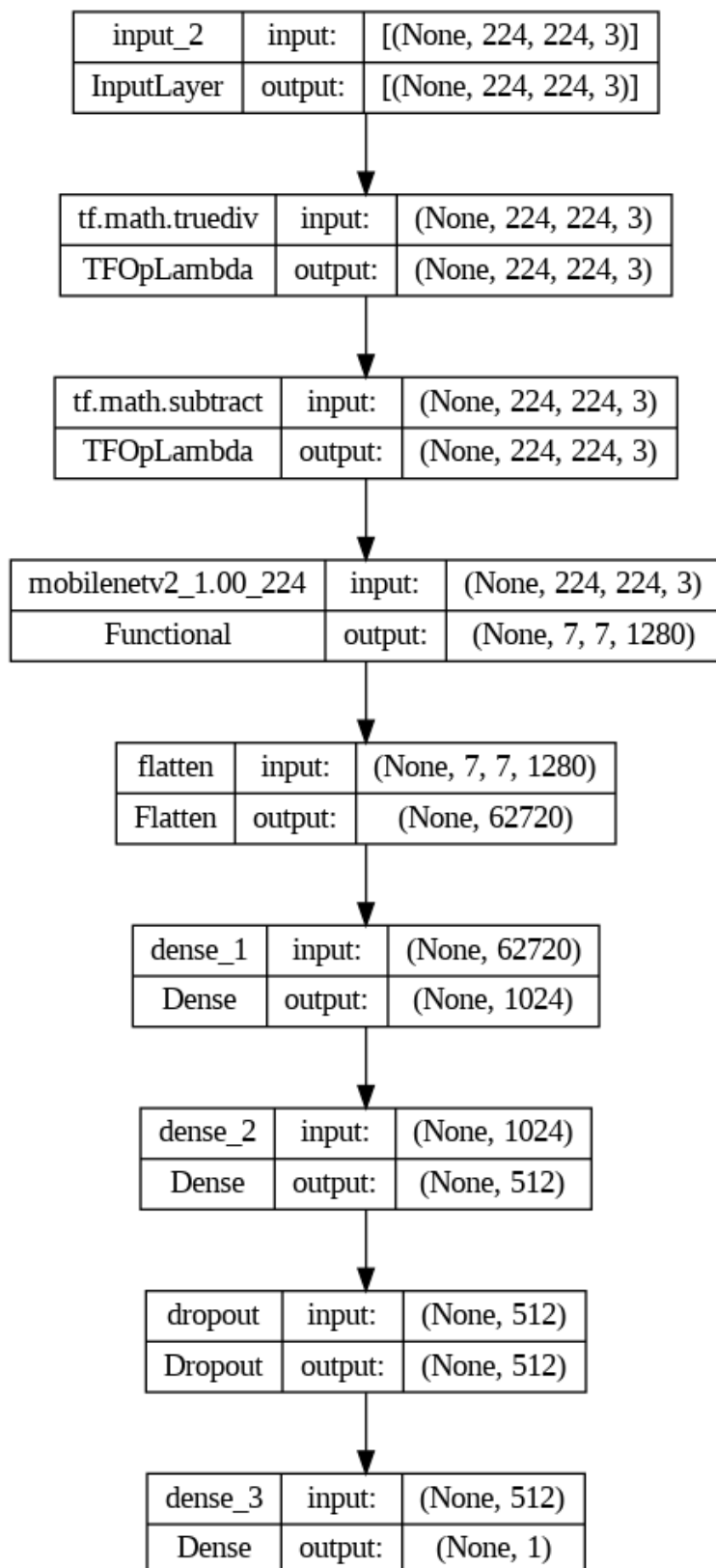
```
inputs = tf.keras.Input(shape=(224, 224, 3))
x = data_augmentation(inputs)
x = preprocess_input(inputs)
x = base_model(x, training=False)
x = tf.keras.layers.Flatten()(x)
x = Dense(1024)(x)
x = Dense(512)(x)
x = tf.keras.layers.Dropout(0.4)(x)
outputs = Dense(1)(x)
model = tf.keras.Model(inputs, outputs)
```

Εικόνα 3.13. Η τελική αρχιτεκτονική του μοντέλου βασισμένο το mobilenetv2 μοντέλο.

```
len(model.trainable_variables)
```

```
6
```

Παρατηρούμε ότι έχουμε 6 αντικείμενα που μπορούν να εκπαιδευτούν τα οποία είναι τα βάρη και οι πολώσεις για τα τρία πλήρως συνδεδεμένα στρώματα που προστέθηκαν στην κεφαλή κατηγοριοποίησης.



Εικόνα 3.14. Γραφική αναπαράσταση της αρχιτεκτονικής του μοντέλου MobileNet2.

3.5 - Εκπαίδευση μοντέλου

Μια συνάρτηση απώλειας μετρά πόσο το μοντέλο αποκλίνει από τη σωστή πρόβλεψη. Οι συναρτήσεις απώλειας παρέχουν περισσότερα από μια στατική απεικόνιση του πόσο καλά λειτουργεί το μοντέλο αλλά χρησιμεύουν επίσης ως βάση για το πόσο ακριβείς ταιριάζουν τα αποτελέσματα του μοντέλου στα πραγματικά δεδομένα. Οι περισσότεροι αλγόριθμοι βαθιάς μάθησης χρησιμοποιούν μια συνάρτηση απώλειας κατά τη φάση βελτιστοποίησης, η οποία μας βοηθά στην επιλογή των βέλτιστων παραμέτρων (βάρια) των δεδομένων.

Η εξίσωση σφάλματος που θα χρησιμοποιηθεί είναι η cross entropy για δυο κλάσεις (binary_crossentropy).

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$

Εικόνα 3.15. Εξίσωση σφάλματος binary cross-entropy [36].

Η συνάρτηση βελτιστοποίησης που θα χρησιμοποιηθεί είναι ο Adam, εισήχθη το 2015 από τους Diederik Kingma & Jimmy Ba και είναι ένας αλγόριθμος διαφορικού πρώτης τάξης για στοχαστικές συναρτήσεις, με βάση προσαρμοσμένων εκτιμήσεων ροπών χαμηλής τάξης [26]. Ο ρυθμός μάθησης, learning rate, είναι ο προκαθορισμένος δηλαδή 0.0001.

Θα χρησιμοποιήσουμε τμήματα δεδομένων με μέγεθος (batch size) ίσο με 10 για 10 εποχές, επαναλήψεις εκπαίδευσης, με πρόωρο σταμάτημα (Early stopping) όταν το μοντέλο έχει ακρίβεια εκπαίδευσης 99.5%.

```
class myCallback(tf.keras.callbacks.Callback):
    # Define the correct function signature for on_epoch_end
    def on_epoch_end(self, epoch, logs={}):
        if logs.get('accuracy') is not None and logs.get('accuracy') > 0.995:
            print("\nReached 99.5% accuracy so cancelling training!")

            # Stop training once the above condition is met
            self.model.stop_training = True

callbacks = myCallback()
```

```
base_learning_rate = 0.0001
model.compile(optimizer=Adam(learning_rate=base_learning_rate),
              loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
              metrics=[tf.keras.metrics.BinaryAccuracy(threshold=0, name='accuracy')])
```

Εικόνα 3.16. Κώδικας για την κατασκευή του custom callback και compile του μοντέλου με τις επιθυμητές υπερπαραμέτρους.

Το μοντέλο αρχικά θα εκπαιδευτεί για δέκα εποχές, παρατηρούμε επίσης ότι το μοντέλο ξεκινάει με ακρίβεια στα δεδομένα ελέγχου στο 28%.

```
loss0, accuracy0 = model.evaluate(test_dataset)

4/4 [=====] - 3s 165ms/step - loss: 1.6743 - accuracy: 0.2778

print("initial loss: {:.2f}".format(loss0))
print("initial accuracy: {:.2f}".format(accuracy0))

initial loss: 1.67
initial accuracy: 0.28
```

Εικόνα 3.17. Κώδικας αρχικοποίηση υπερ-παραμέτρων.

Παρατηρούμε ότι κατά την διάρκεια της εκπαίδευσης το σφάλμα μειώνεται και η ακρίβεια αυξάνεται που είναι αναμενόμενο και δείχνει ότι το μοντέλο μαθαίνει τα δεδομένα.

```
Epoch 9/10
39/39 [=====] - 8s 187ms/step - loss: 1.1040 - accuracy: 0.9306 - val_loss: 2.4532 - val_accuracy: 0.8056
Epoch 10/10
39/39 [=====] - 7s 153ms/step - loss: 0.7681 - accuracy: 0.9434 - val_loss: 1.9890 - val_accuracy: 0.8056
```

Μετά 10 απόχες έχει αυξηθεί η ακρίβεια στα δεδομένα εκπαίδευσης και στα δεδομένα ελέγχου παρόλα αυτά όμως υπάρχει μεγάλη διαφορά μεταξύ της ακρίβειας τα δεδομένα εκπαίδευσης και στα δεδομένα ελέγχου και η ακρίβεια στα δεδομένα ελέγχου είναι ακόμα χαμηλή, αυτό θα αντιμετωπιστεί με το fine tuning του μοντέλου.


```
def graph(history, model):
    acc = history.history['accuracy']
    test_acc = history.history['val_accuracy']

    loss = history.history['loss']
    test_loss = history.history['val_loss']

    plt.figure(figsize=(8, 8))
    plt.subplot(2, 1, 1)
    plt.plot(acc, label='Training Accuracy')
    plt.plot(test_acc, label='Validation Accuracy')
    plt.legend(loc='lower right')
    plt.ylabel('Accuracy')
    plt.ylim([min(plt.ylim()),1])
    plt.title('Training and Validation Accuracy for ' + model)

    plt.subplot(2, 1, 2)
    plt.plot(loss, label='Training Loss')
    plt.plot(test_loss, label='Validation Loss')
    plt.legend(loc='upper right')
    plt.ylabel('Cross Entropy')
    plt.ylim([0, 10.0])
    plt.title('Training and Validation Loss for ' + model)
    plt.xlabel('epoch')
    plt.show()

graph(history, 'MobileNetV2')
```

Εικόνα 3.18. Κώδικας γραφικοποίηση της ακρίβειας και του σφάλματος για τα δεδομένα ελέγχου και εκπαίδευση ανά εποχή.



Εικόνα 3.19. Διάγραμμα ακρίβειας, σφάλματος- εποχής για τα δεδομένα εκπαίδευσης και ελέγχου για το mobilenetv2.

3.5.1 Fine Tuning

Θα πραγματοποιηθεί fine tuning των βασικών μοντέλων με τα δεδομένα μας για να αυξήσουμε την απόδοση των μοντέλων.

Αρχικά , ξεπαγώνουμε το βασικό μοντέλο και ορίζουμε σε ποιά στρώματα θα κάνουμε εκπαίδευση με τα δεδομένα μας, fine tuning,ότι δεν θα εκπαιδευτεί θα τα παγώσουμε.

```
base_model.trainable = True

# Let's take a look to see how many layers are in the base model
print("Number of layers in the base model: ", len(base_model.layers))

# Fine-tune from this layer onwards
fine_tune_at = 100

# Freeze all the layers before the `fine_tune_at` layer
for layer in base_model.layers[:fine_tune_at]:
    layer.trainable = False

Number of layers in the base model: 154
```

Θα χρησιμοποιήσουμε ένα μικρότερο learning rate ώστε το μοντέλο να εκπαιδευτεί πιο αργά ώστε να αποφύγουμε την υπέρεκπαίδευση αφού έχουμε μικρό dataset.

```
model.compile(loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
              optimizer = tf.keras.optimizers.RMSprop(learning_rate=base_learning_rate/10),
              metrics=[tf.keras.metrics.BinaryAccuracy(threshold=0, name='accuracy')])
```

Παρατηρούμε ότι έχει αυξηθεί το πλήθος των αντικειμένων που μπορεί να εκπαιδευτούν.

```
len(model.trainable_variables)

60
```

Θα συνεχίσουμε την εκπαίδευση εκεί που την είχαμε αφήσει για ακόμα 10 εποχές.

```
fine_tune_epochs = 10
total_epochs = initial_epochs + fine_tune_epochs

history_fine = model.fit(train_dataset,
                        epochs=total_epochs,
                        initial_epoch=history.epoch[-1],
                        validation_data=test_dataset,
                        callbacks=[callbacks])
```

Παρατηρούμε ότι το μοντέλο κατάφερε να φτάσει ακρίβεια 97% στα δεδομένα εκπαίδευσης και βλέπουμε ότι έχει αυξηθεί σημαντικά και η ακρίβεια στα δεδομένα ελέγχου.

```
Epoch 19/20
39/39 [=====] - 9s 187ms/step - loss: 0.0966 - accuracy: 0.9846 - val_loss: 1.6766 - val_accuracy: 0.8333
Epoch 20/20
39/39 [=====] - 8s 187ms/step - loss: 0.1094 - accuracy: 0.9717 - val_loss: 1.5648 - val_accuracy: 0.8611
```

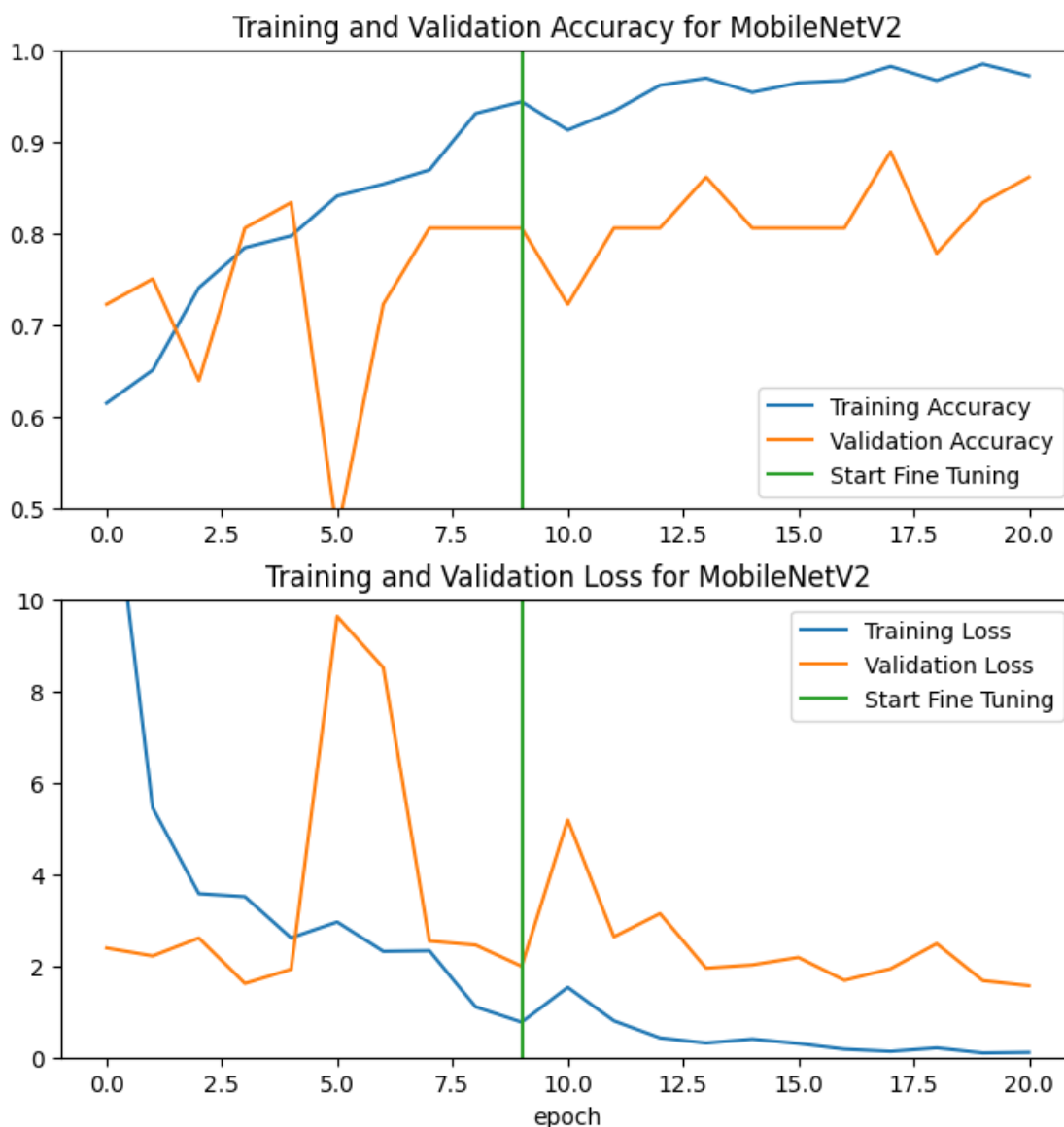
```
def fine_tuned_graph(history_fine, acc, test_acc, loss, test_loss, model):
    acc += history_fine.history['accuracy']
    test_acc += history_fine.history['val_accuracy']

    loss += history_fine.history['loss']
    test_loss += history_fine.history['val_loss']

    plt.figure(figsize=(8, 8))
    plt.subplot(2, 1, 1)
    plt.plot(acc, label='Training Accuracy')
    plt.plot(test_acc, label='Validation Accuracy')
    plt.ylim([0.5, 1])
    plt.plot([initial_epochs-1,initial_epochs-1],
             plt.ylim(), label='Start Fine Tuning')
    plt.legend(loc='lower right')
    plt.title('Training and Validation Accuracy for ' + model)

    plt.subplot(2, 1, 2)
    plt.plot(loss, label='Training Loss')
    plt.plot(test_loss, label='Validation Loss')
    plt.ylim([0, 10.0])
    plt.plot([initial_epochs-1,initial_epochs-1],
             plt.ylim(), label='Start Fine Tuning')
    plt.legend(loc='upper right')
    plt.title('Training and Validation Loss for ' + model)
    plt.xlabel('epoch')
    plt.show()
```

Εικόνα 3.20. Κώδικας για την γραφικόποίηση των αποτελεσμάτων πριν και μετά του fine tuning.



Εικόνα 3.21. Γραφική αναπαράσταση της ακρίβειας και του σφάλματος για τα δεδομένα εκπαίδευσης και ελέγχου ανά εποχή πριν και μετά το fine tuning για το μοντέλο MobileNet2.

3.6 - Εξαγωγή συμπερασμάτων

Κατά την εξαγωγή των συμπερασμάτων θέλουμε όχι μόνο να ταξινομούμε μία εικόνα στις κατηγορίες Good και Anomaly αλλά θέλουμε και ταυτόχρονα να παίρνουμε και ένα κουτί χαρακτηριστικών που θα μας δείχνει πού είναι το ελαττωματικό σημείο της εικόνας.

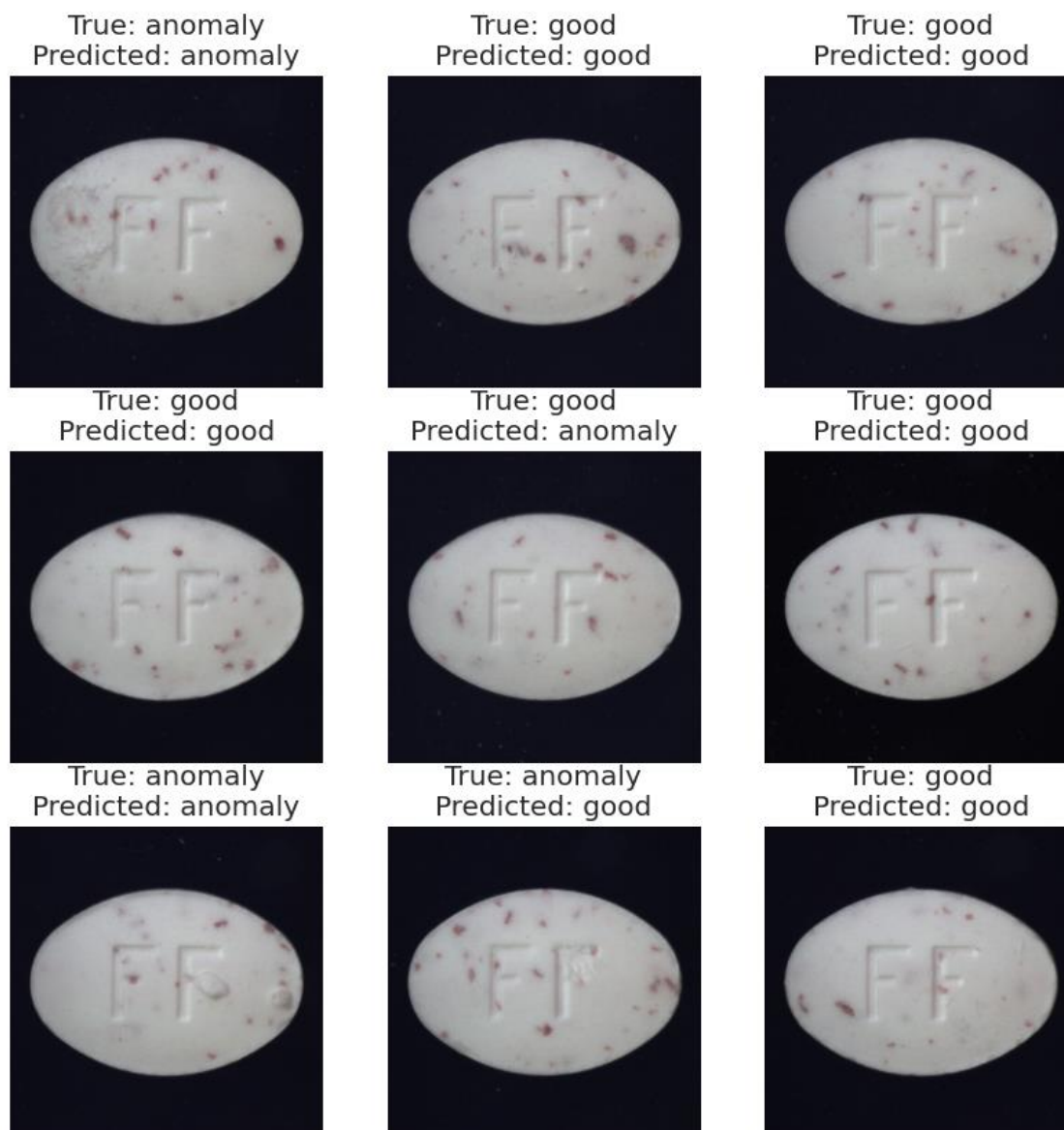
```
# Retrieve a batch of images from the test set
image_batch, label_batch = test_dataset.as_numpy_iterator().next()
predictions = model.predict_on_batch(image_batch).flatten()

# Apply a sigmoid since our model returns logits
predictions = tf.nn.sigmoid(predictions)
predictions = tf.where(predictions < 0.5, 0, 1)

print('Predictions:\n', predictions.numpy())
print('Labels:\n', label_batch)

plt.figure(figsize=(10, 10))
for i in range(9):
    ax = plt.subplot(3, 3, i + 1)
    plt.imshow(image_batch[i].astype("uint8"))
    true_label = class_names[label_batch[i]] # Get the true label
    predicted_label = class_names[predictions[i]] # Get the predicted label
    title = f"True: {true_label}\nPredicted: {predicted_label}"
    plt.title(title)
    plt.axis("off")
```

Εικόνα 3.22. Κώδικας για τον έλεγχο της ικανότητας πρόβλεψης σωστών κατηγοριών του μοντέλου σε ένα batch των δεδομένων ελέγχου.



Εικόνα 3.23. Γραφική απεικόνιση 9 εικόνων από το σύνολο ελέγχου.

3.6.1 - Στατιστικά στοιχεία

Μπορούμε τελικά να εξάγουμε ένα πίνακα σύγχυσης (confusion matrix) που θα μας δίνει τα πραγματικά θετικά (True Positive), πραγματικά αρνητικά (True Negative), τα εσφαλμένα θετικά (False Positives), εσφαλμένα αρνητικά (False Negatives) που μπορεί να μας δώσει απαραίτητα στατιστικά στοιχεία για την επίδοση του μοντέλου στα δεδομένα ελέγχου.

Ακρίβεια (accuracy):

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \quad (4)$$

Η **ισορροπημένη ακρίβεια (balanced accuracy)** σε προβλήματα ταξινόμησης δυαδικών και πολλαπλών κλάσεων για την αντιμετώπιση μη ισορροπημένων συνόλων δεδομένων. Ορίζεται ως ο μέσος όρος ανάκλησης που λαμβάνεται σε κάθε τάξη. Η καλύτερη τιμή είναι 1 και η χειρότερη τιμή είναι 0.

$$Balance Accuracy = \frac{sensitivity + specificity}{2} \quad (5)$$

όπου:

Ευαισθησία (Sensitivity) επίσης γνωστό ως αληθινό θετικό ποσοστό ή ανάκληση, μετρά την αναλογία των πραγματικών θετικών που έχουν προβλεφθεί σωστά από όλες τις θετικές προβλέψεις που θα μπορούσαν να γίνουν από το μοντέλο.

$$Sensitivity = \frac{TP}{TP + FN} \quad (6)$$

Ιδιαιτερότητα (Specificity) επίσης γνωστό και ως πραγματικό αρνητικό ποσοστό, μετρά την αναλογία των σωστά προσδιορισμένων αρνητικών σε σχέση με τις συνολικές αρνητικές προβλέψεις που θα μπορούσαν να γίνουν από το μοντέλο.

$$Specificity = \frac{TN}{TN + FP} \quad (7)$$

Ευστοχία (Precision): Η ακρίβεια είναι διαισθητικά η ικανότητα του ταξινομητή να μην επισημαίνει ως θετικό ένα δείγμα που είναι αρνητικό. Η καλύτερη τιμή είναι 1 και η χειρότερη τιμή είναι 0.

$$Precision = \frac{TP}{POSITIVE} = \frac{TP}{TP + FP} \quad (8)$$

Ανάκληση (Recall): Η ανάκληση είναι διαισθητικά η ικανότητα του ταξινομητή να βρίσκει όλα τα θετικά δείγματα. Η καλύτερη τιμή είναι 1 και η χειρότερη τιμή είναι 0.

$$Recall = \frac{TP}{Class 1} = \frac{TP}{TP + FN} \quad (9)$$

F1-score: Η βαθμολογία F1 μπορεί να ερμηνευθεί ως ένας αρμονικός μέσος όρος της ακρίβειας και της ανάκλησης, όπου μια βαθμολογία F1 φτάνει την καλύτερη τιμή της στο 1 και τη χειρότερη βαθμολογία στο 0. Η σχετική συμβολή της ακρίβειας και της ανάκλησης στη βαθμολογία της F1 είναι ίση.

$$F1\ measure = \frac{Precision \cdot Recall}{(Precision + Recall)/2} \quad (10)$$

macro avg:

Υπολογίζει τις μετρήσεις για κάθε ετικέτα και βρίσκει τον μη σταθμισμένο μέσο όρο τους. Αυτό δεν λαμβάνει υπόψη την ανισορροπία της ετικέτας.

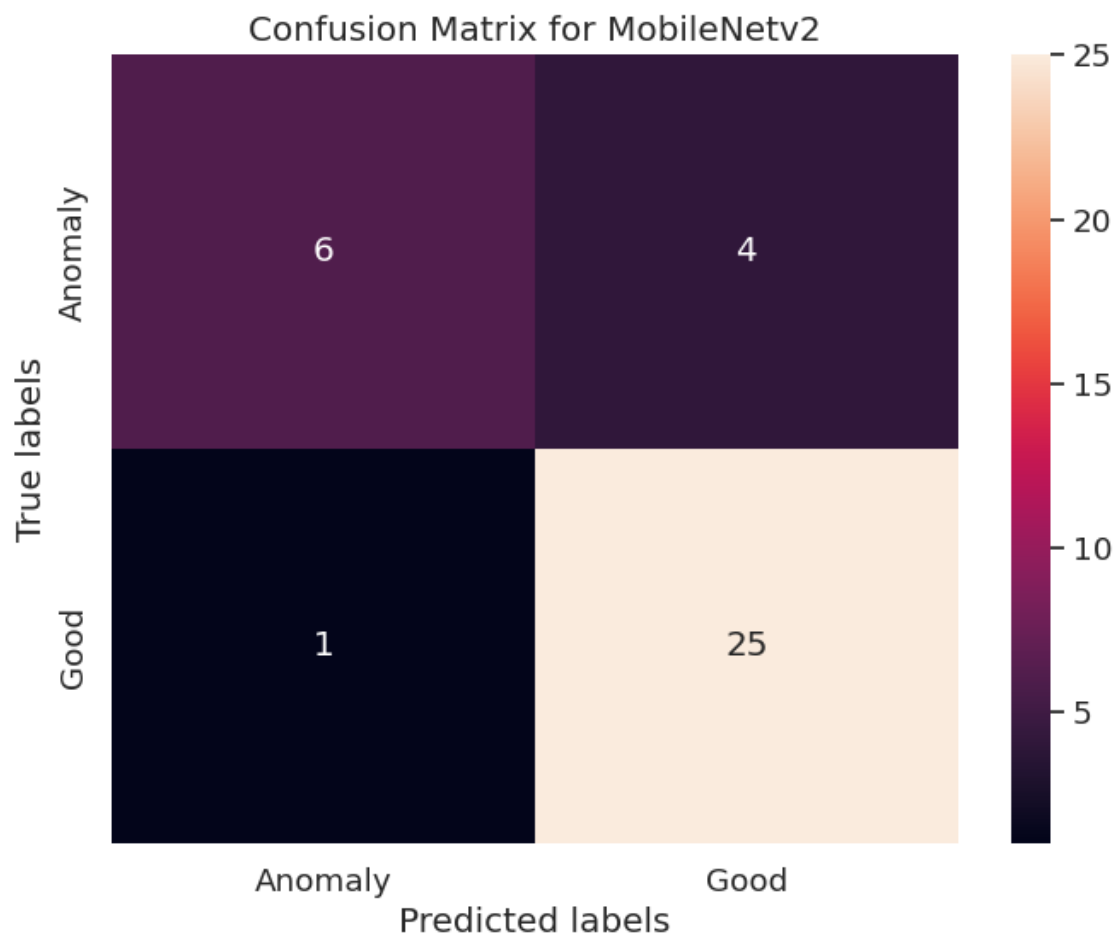
weighted avg:

Υπολογίζει τις μετρήσεις για κάθε ετικέτα και βρίσκει τη μέση στάθμη τους με βάση την υποστήριξη (ο αριθμός των πραγματικών περιπτώσεων για κάθε ετικέτα). Αυτό έχει ως αποτέλεσμα για να ληφθεί υπόψη η ανισορροπία των ετικετών.

Για την εξαγωγή στατιστικών κατηγοριοποίησης χρησιμοποιήσουμε τη βιβλιοθήκη sklearn και seaborn. Παίρνουμε τα αποτελέσματα τις προβλέψεις του μοντέλου στα επόμενα ελέγχου τα μετατρέπουμε σε τιμές 0, 1 τα συγκρίνουμε με τις αληθινές κατηγορίες ώστε να φτιάξουμε την μήτρα σύγκρισης και τον πίνακα κατηγοριοποίησης.

```
from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sns

y_pred = model.predict(test_dataset)
print(y_pred)
preds = tf.nn.sigmoid(y_pred.flatten())
predicted_categories = tf.where(preds < 0.5, 0, 1)
print(predicted_categories)
true_categories = tf.concat([y for x, y in test_dataset], axis=0)
print(true_categories)
confusion_mat = confusion_matrix(true_categories, predicted_categories)
print(confusion_mat)
```



Εικόνα 3.24. Μήτρα σύγχυσης για mobilenetv2.

	Precision	Recall	F1-score	Support
Anomaly	0.857	0.600	0.709	10
Good	0.862	0.962	0.909	26
Accuracy			0.861	36
Macro Avg	0.860	0.781	0.807	36
Weighted Avg	0.861	0.861	0.853	36

Πίνακας 3.1. Στατιστικά στοιχεία ταξινόμησης για mobilenetv2.

3.7 - Εκπαίδευση με άλλα δημοφιλή μοντέλα και σύγκριση για την εύρεση του καλύτερου.

Υπάρχει μεγάλη ποικιλία από μοντέλα υπολογιστικής όρασης που χρησιμοποιούνται συχνά για ταξινόμηση εικόνων, το καθένα μοναδικό ως προς την δομή, την αρχιτεκτονική και το μέγεθος.

Τα μοντέλα θα εκπαιδευτούν και θα αξιολογηθούν στο ίδιο σετ δεδομένων εκπαίδευσης και ελέγχου με τις ίδιες υπερπαραμέτρους ώστε να βρεθεί αυτό με την καλύτερη απόδοση.

Ομοίως όπως έχει περιγραφεί παραπάνω θα ακολουθήσουμε τα βήματα για τα άλλα μοντέλα δηλαδή:

- Κατεβάζουμε τα μοντέλα τα παγώνουμε και αφαιρούμε την κεφαλή κατηγοριοποίηση τους.

```

Ας δούμε πως συγκρίνεται η απόδοση του MobileNet V2 με άλλα δημοφιλή μοντέλα.

[ ] # Define the pre-trained models you want to compare
base_model1 = VGG16(input_shape=IMG_SHAPE, weights='imagenet', include_top=False)
base_model2 = ResNet50(input_shape=IMG_SHAPE, weights='imagenet', include_top=False)
base_model3 = InceptionV3(input_shape=IMG_SHAPE, weights='imagenet', include_top=False)
base_model4 = EfficientNetB0(input_shape=IMG_SHAPE, weights='imagenet', include_top=False)
base_model5 = VGG19(input_shape=IMG_SHAPE, weights='imagenet', include_top=False)

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58889256/58889256 [=====] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5
94765736/94765736 [=====] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_v3/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5
87910968/87910968 [=====] - 0s 0us/step
Downloading data from https://storage.googleapis.com/keras-applications/efficientnetb0_notop.h5
16705208/16705208 [=====] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg19/vgg19_weights_tf_dim_ordering_tf_kernels_notop.h5
88134624/88134624 [=====] - 0s 0us/step

```

- Χρησιμοποιούμε τα preprocessing player που έρχονται από το tf.keras.applications ώστε να έχουμε την κατάλληλη μορφή των εικόνων κατά την είσοδο τους στο μοντέλο.

```

preprocess_input1 = tf.keras.applications.vgg16.preprocess_input
preprocess_input2 = tf.keras.applications.resnet50.preprocess_input
preprocess_input3 = tf.keras.applications.inception_v3.preprocess_input
preprocess_input4 = tf.keras.applications.efficientnet.preprocess_input
preprocess_input5 = tf.keras.applications.vgg19.preprocess_input

```

```

image_batch1, label_batch1 = next(iter(train_dataset))
feature_batch1 = base_model1(image_batch1)
print(feature_batch1.shape)

(10, 7, 7, 512)

```

Εικόνα 3.25. α) Ο χάρτης χαρακτηριστικών για το μοντέλο vgg16.

```
image_batch2, label_batch2 = next(iter(train_dataset))
feature_batch2 = base_model2(image_batch2)
print(feature_batch2.shape)

(10, 7, 7, 2048)
```

Εικόνα 3.25. β) Ο χάρτης χαρακτηριστικών για το μοντέλο resnet50.

```
image_batch3, label_batch3 = next(iter(train_dataset))
feature_batch3 = base_model3(image_batch3)
print(feature_batch3.shape)

(10, 5, 5, 2048)
```

Εικόνα 3.25. γ) Ο χάρτης χαρακτηριστικών για το μοντέλο Inception v3.

```
image_batch4, label_batch4 = next(iter(train_dataset))
feature_batch4 = base_model4(image_batch4)
print(feature_batch4.shape)

(10, 7, 7, 1280)
```

Εικόνα 3.25. δ) Ο χάρτης χαρακτηριστικών για το μοντέλο EfficientNetb0.

```
image_batch5, label_batch5 = next(iter(train_dataset))
feature_batch5 = base_model5(image_batch5)
print(feature_batch5.shape)

(10, 7, 7, 512)
```

Εικόνα 3.25. ε) Ο χάρτης χαρακτηριστικών για το μοντέλο vgg19.

- Παγώνουμε τα βάρια των βασικών μοντέλων.

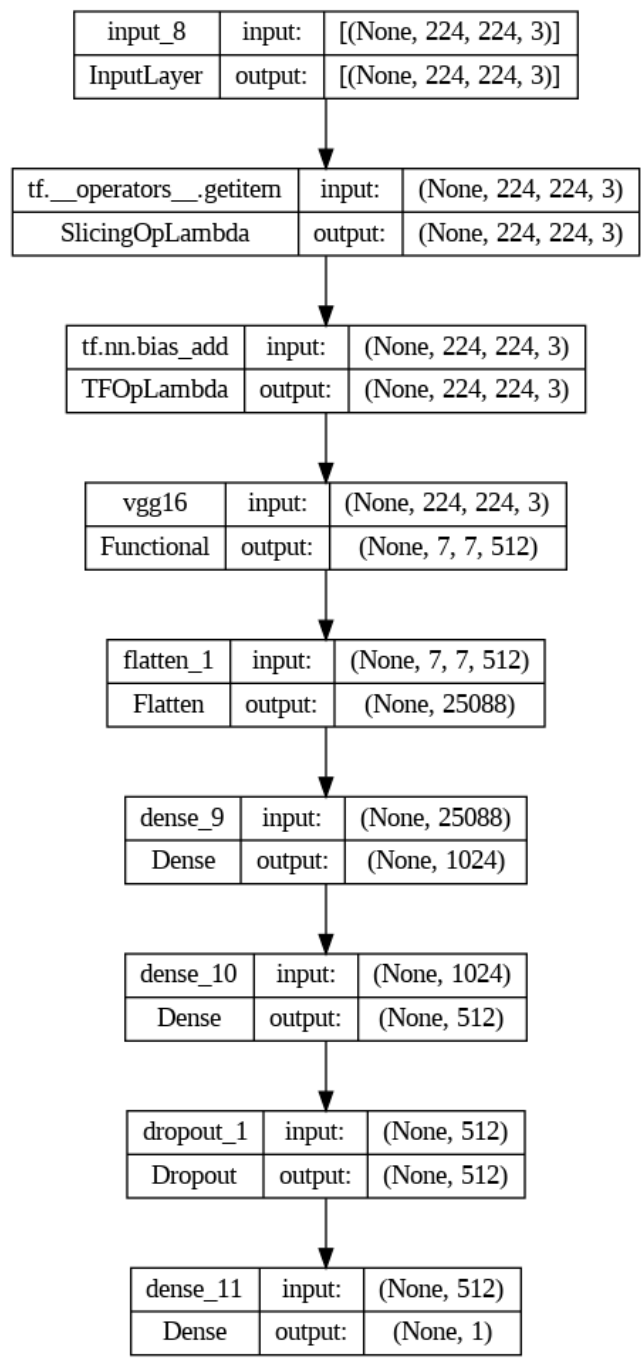
```
base_model1.trainable = False
base_model2.trainable = False
base_model3.trainable = False
base_model4.trainable = False
base_model5.trainable = False
```

- Χρησιμοποιούμε όπως και πριν την ίδια αρχιτεκτονική γύρω από τα βασικά μοντέλα.

```

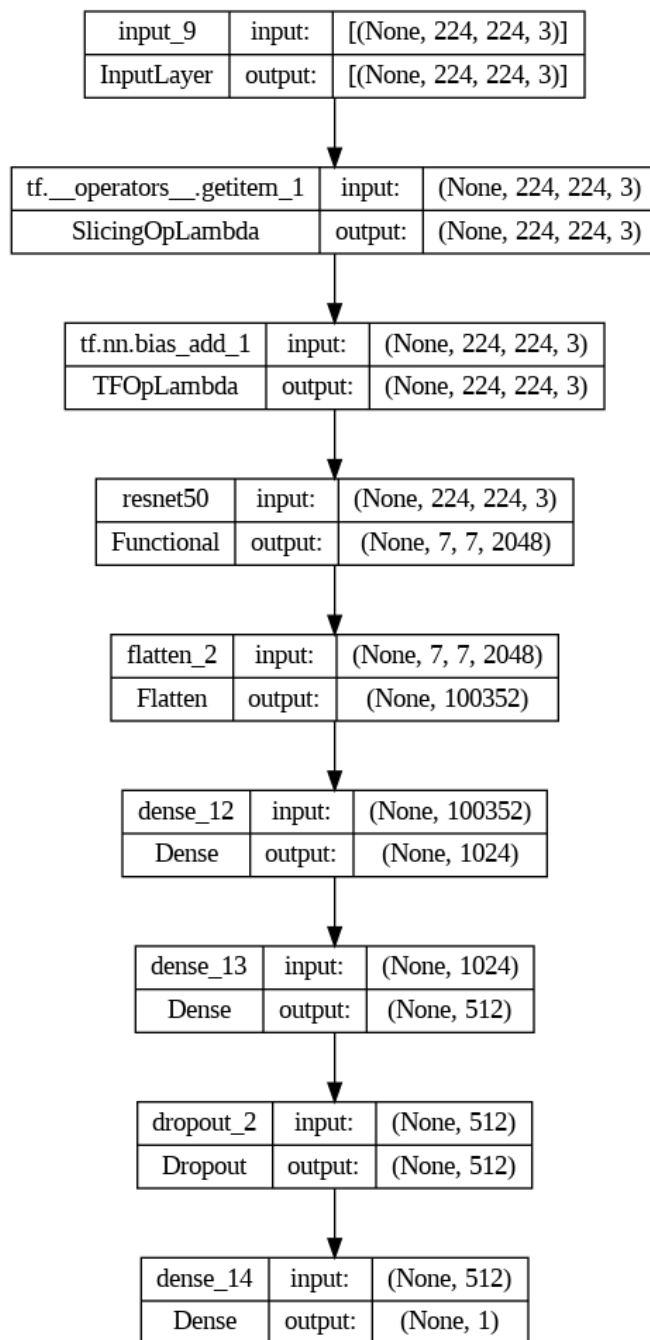
inputs = tf.keras.Input(shape=(224, 224, 3))
x = data_augmentation(inputs)
x = preprocess_input(inputs)
x = base_model(x, training=False)
x = tf.keras.layers.Flatten()(x)
x = Dense(1024)(x)
x = Dense(512)(x)
x = tf.keras.layers.Dropout(0.4)(x)
outputs = Dense(1)(x)
model = tf.keras.Model(inputs, outputs)

```

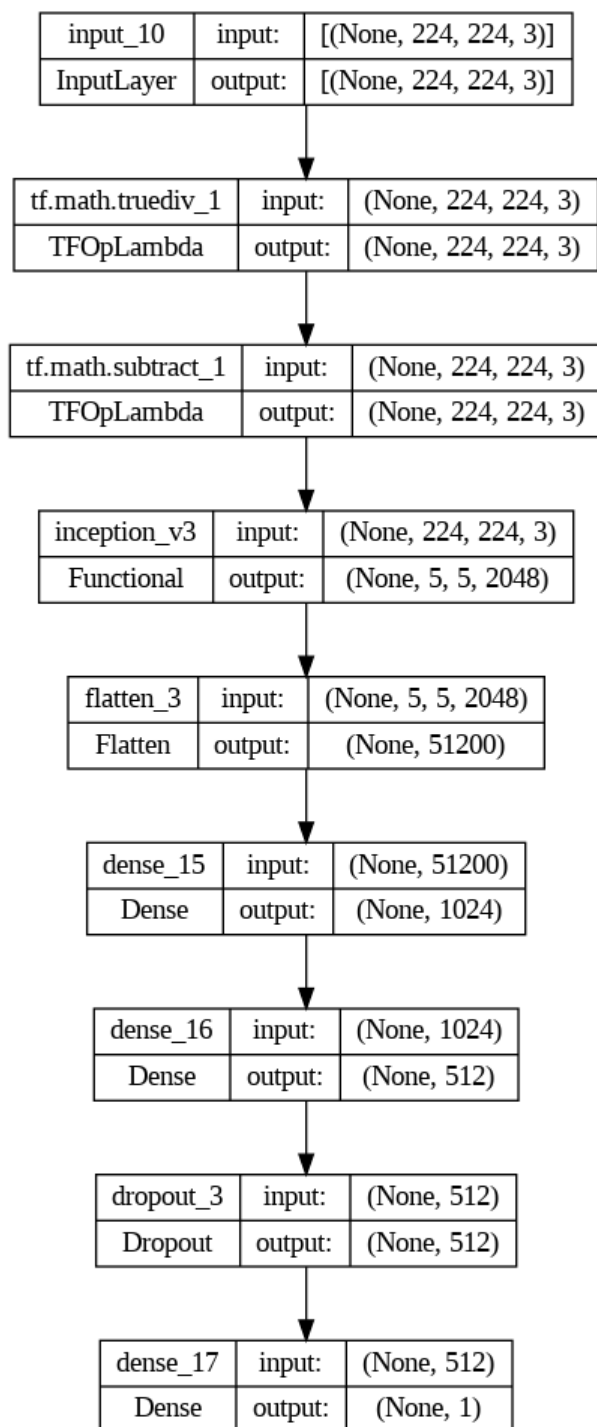


Εικόνα 3.26. α) Γραφική απεικόνιση της αρχιτεκτονικής του μοντέλου με βασικό

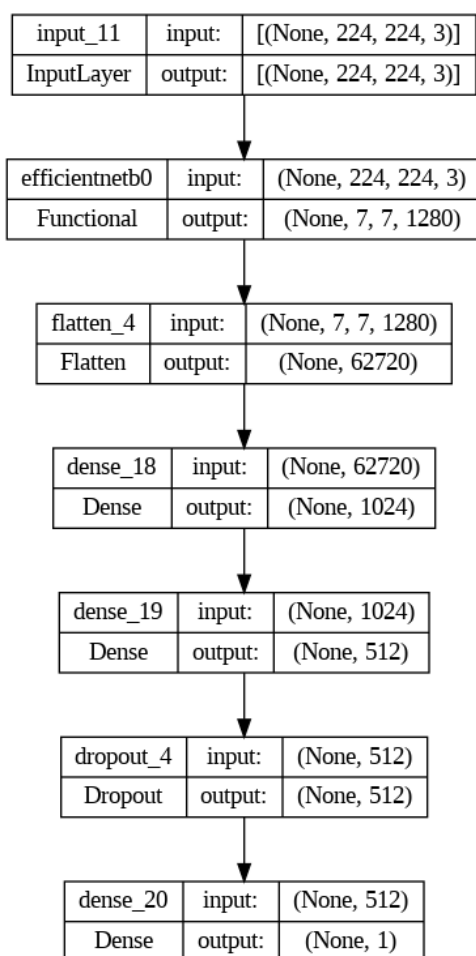
μοντέλο VGG16 με το μέγεθος των εισόδων και των εξόδων του σε κάθε στρώμα.



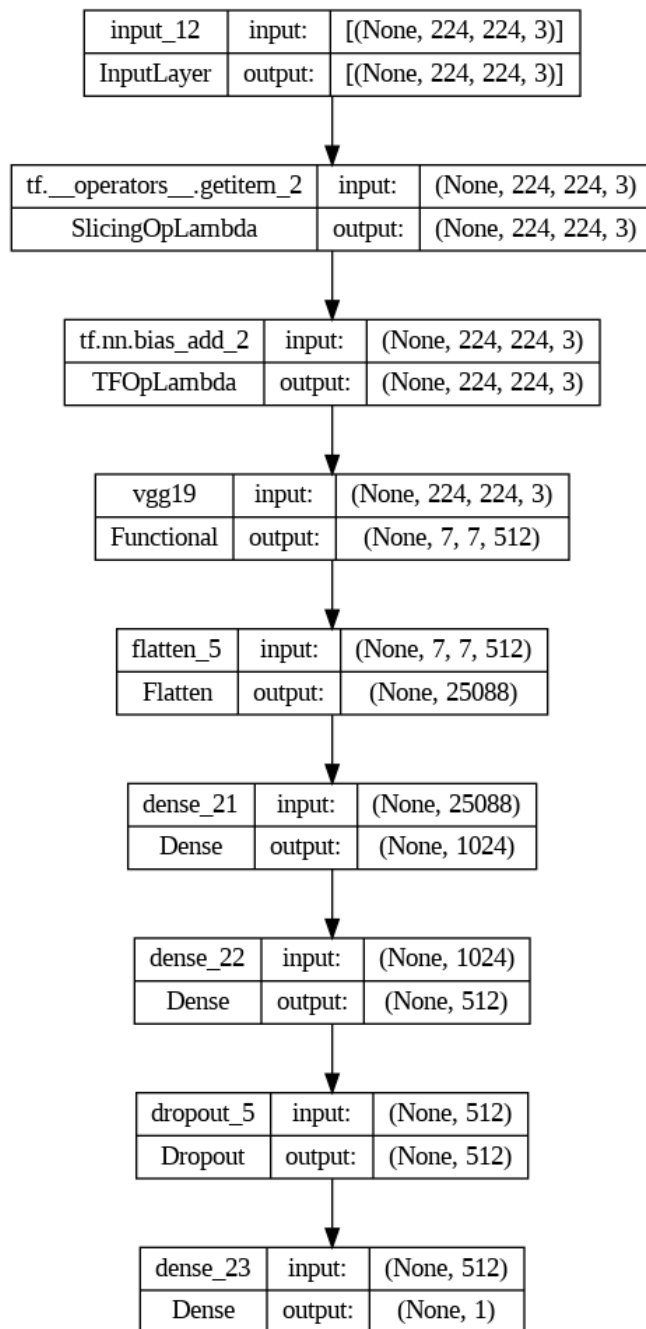
Εικόνα 3.26. β) Γραφική απεικόνιση της αρχιτεκτονικής του μοντέλου με βασικό μοντέλο resnet50 με το μέγεθος των εισόδων και των εξόδων του σε κάθε στρώμα.



Εικόνα 3.26. γ) Γραφική απεικόνιση της αρχιτεκτονικής του μοντέλου με βασικό μοντέλο inceptionv3 με το μέγεθος των εισόδων και των εξόδων του σε κάθε στρώμα.



Εικόνα 3.26. δ) Γραφική απεικόνιση της αρχιτεκτονικής του μοντέλου με βασικό μοντέλο `efficientnetb0` με το μέγεθος των εισόδων και των εξόδων του σε κάθε στρώμα.



Εικόνα 3.26. ε) Γραφική απεικόνιση της αρχιτεκτονικής του μοντέλου με βασικό μοντέλο VGG19 με το μέγεθος των εισόδων και των εξόδων του σε κάθε στρώμα.

Κάνουμε compile το κάθε από αυτά τα μοντέλα με τις ίδιες υπερπαραμέτρους.

```
base_learning_rate = 0.0001
model1.compile(optimizer=Adam(learning_rate=base_learning_rate),
               loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
               metrics=[tf.keras.metrics.BinaryAccuracy(threshold=0, name='accuracy')])
```

Παρατηρούμε ότι όλα ξεκινούν με ακρίβειας τα δεδομένα ελέγχου δεν ξεπερνά το 55%.

```
loss1, accuracy1 = model1.evaluate(test_dataset)
4/4 [=====] - 5s 465ms/step - loss: 2.4495 - accuracy: 0.5278

loss2, accuracy2 = model2.evaluate(test_dataset)
4/4 [=====] - 3s 228ms/step - loss: 4.6546 - accuracy: 0.2778

loss3, accuracy3 = model3.evaluate(test_dataset)
4/4 [=====] - 6s 889ms/step - loss: 1.0224 - accuracy: 0.3889

loss4, accuracy4 = model4.evaluate(test_dataset)
4/4 [=====] - 4s 174ms/step - loss: 2.2474 - accuracy: 0.2778

loss5, accuracy5 = model5.evaluate(test_dataset)
4/4 [=====] - 1s 66ms/step - loss: 1.5970 - accuracy: 0.5556
```

Παρατηρούμε ότι έχουν αυξηθεί σε όλα μοντέλα για ακρίβεια ελέγχου και εκπαίδευσης.

```
loss: 1.3237 - accuracy: 0.9306 - val_loss: 8.4604 - val_accuracy: 0.7778
loss: 0.7544 - accuracy: 0.9614 - val_loss: 1.0701 - val_accuracy: 0.9722
```

α) VGG16

```
loss: 3.5949 - accuracy: 0.8997 - val_loss: 2.2871 - val_accuracy: 0.8611
loss: 2.6172 - accuracy: 0.9152 - val_loss: 5.3216 - val_accuracy: 0.8889
```

β) ResNet50

```
loss: 1.4719 - accuracy: 0.8997 - val_loss: 12.1763 - val_accuracy: 0.5278
loss: 2.2792 - accuracy: 0.8740 - val_loss: 4.5907 - val_accuracy: 0.7500
```

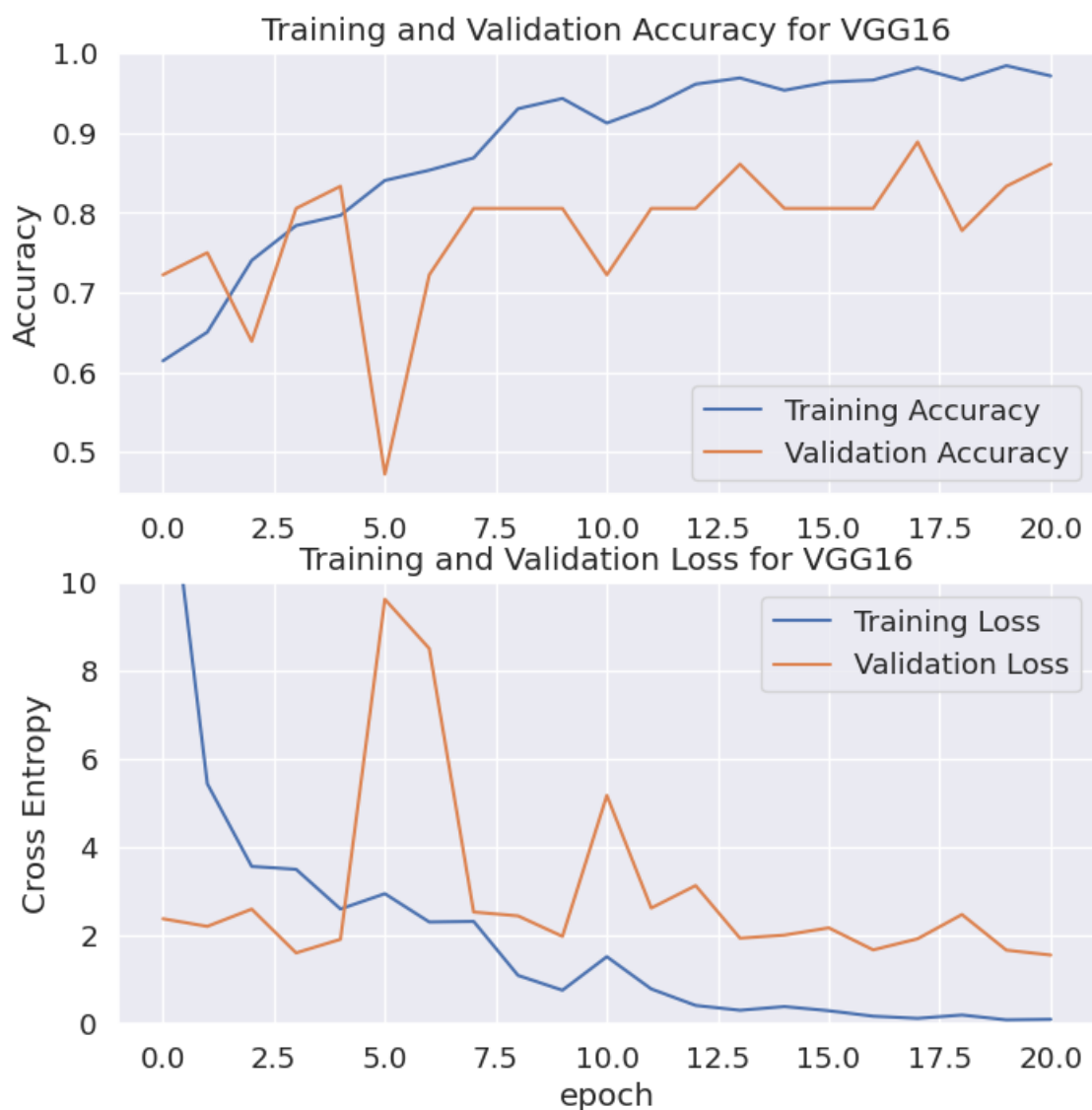
γ) Inceptionv3

```
loss: 1.5135 - accuracy: 0.8920 - val_loss: 3.1697 - val_accuracy: 0.7222
loss: 1.5220 - accuracy: 0.9023 - val_loss: 1.4645 - val_accuracy: 0.8333
```

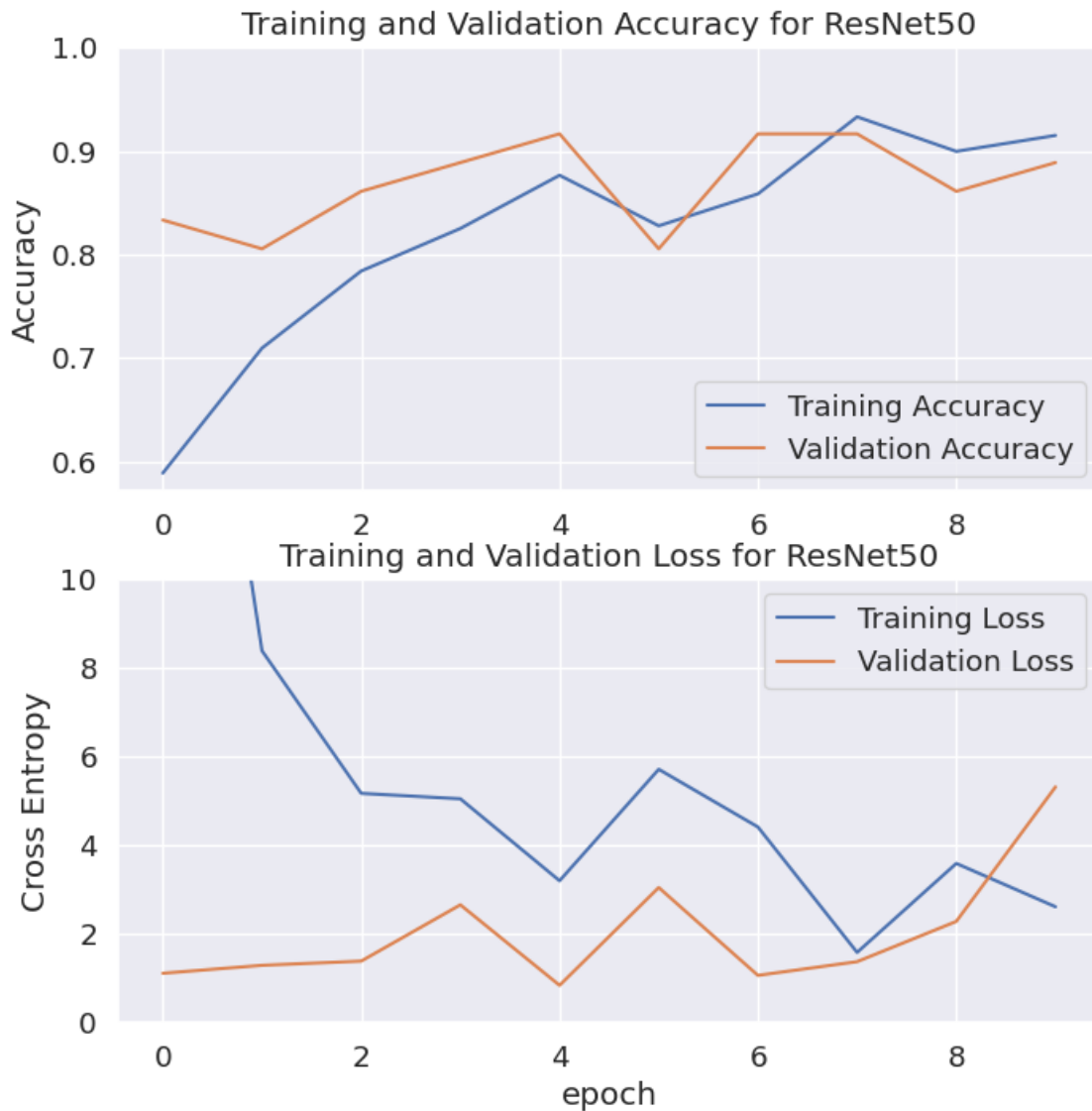
δ) EfficientNetb0

```
loss: 0.5662 - accuracy: 0.9640 - val_loss: 1.5200 - val_accuracy: 0.9167
loss: 0.7069 - accuracy: 0.9537 - val_loss: 6.4780 - val_accuracy: 0.8333
```

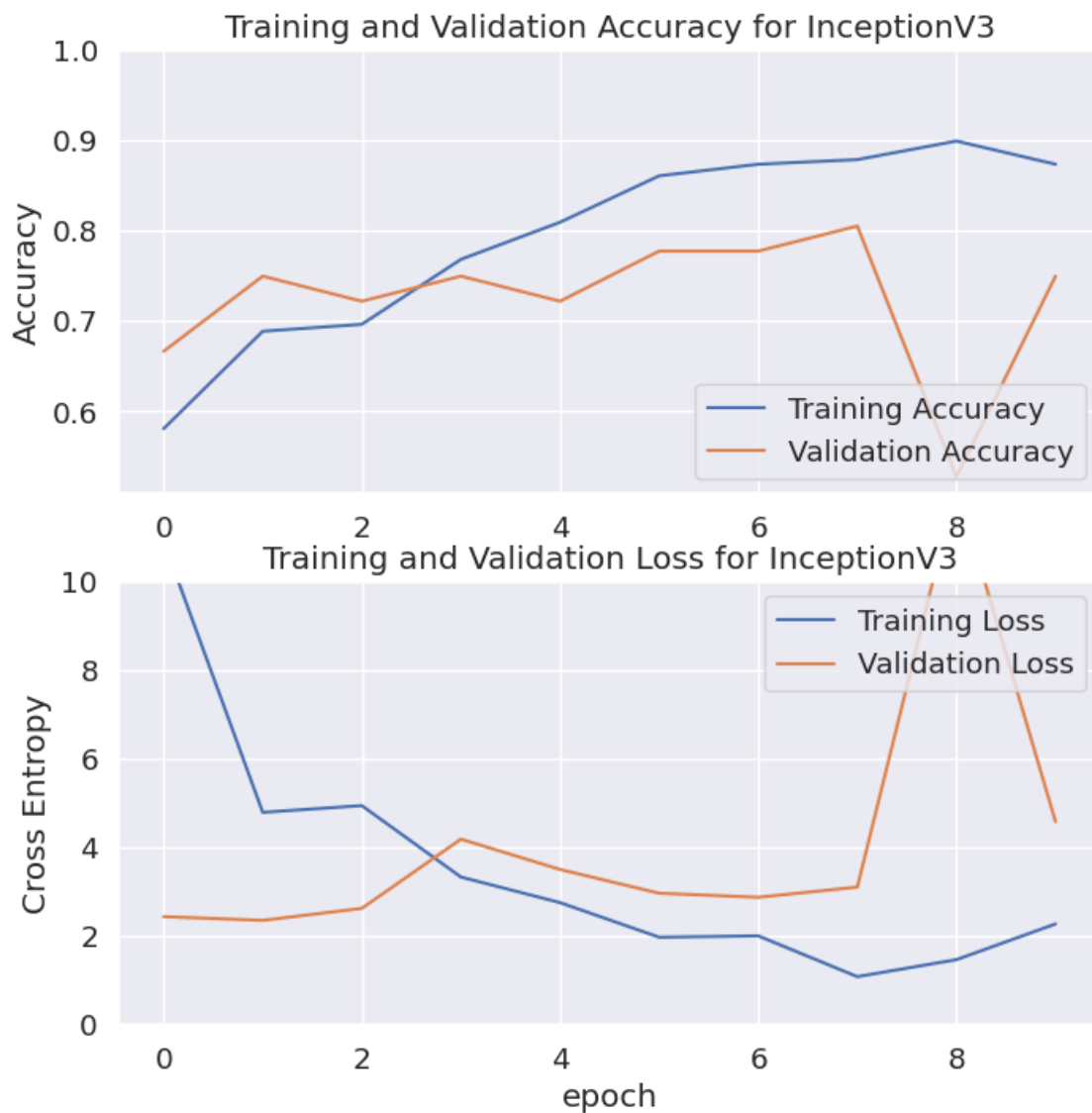
ε) VGG19



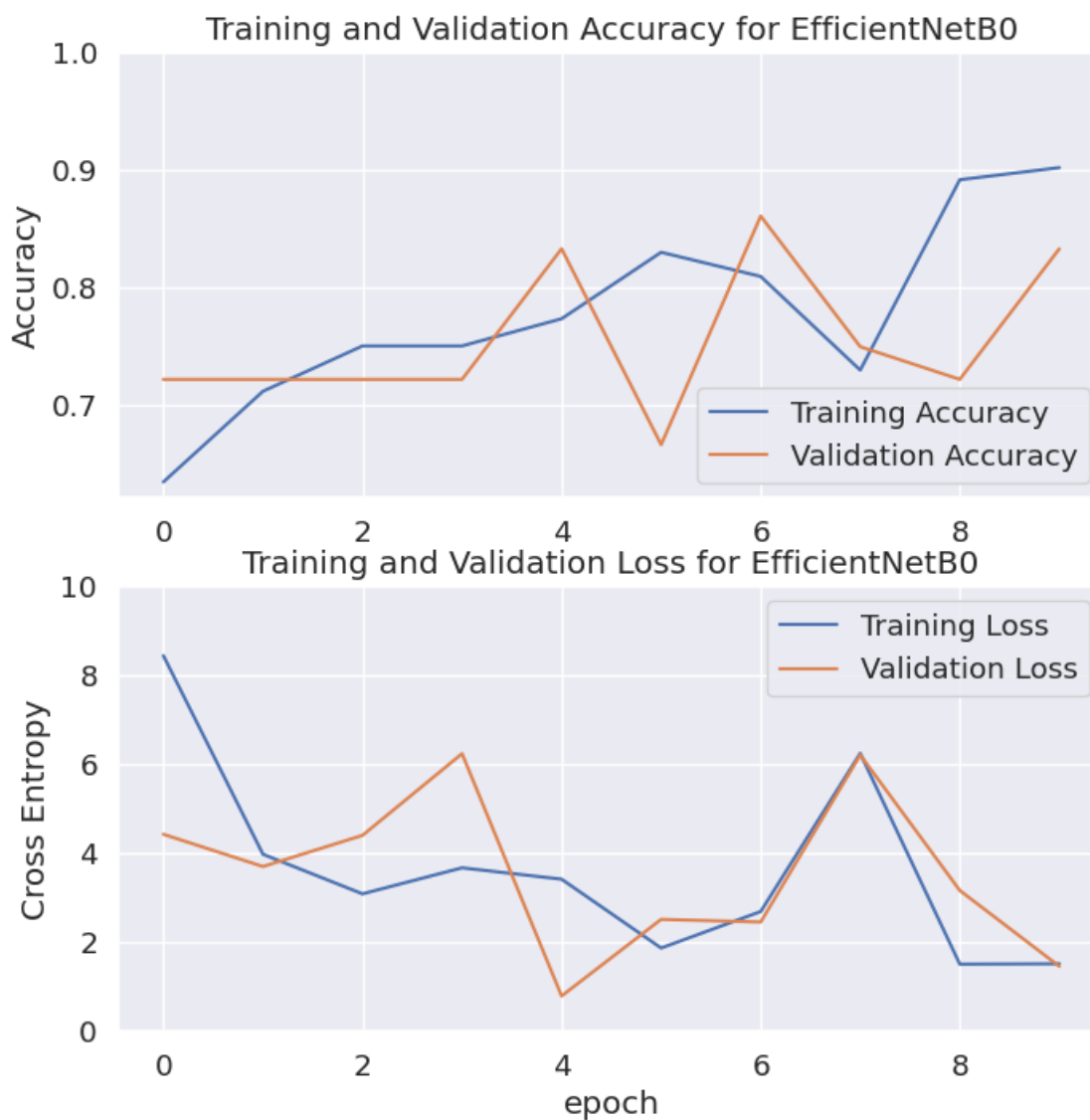
Εικόνα 3.27. α) Γραφική αναπαράσταση της ακρίβειας και του σφάλματος για τα δεδομένα εκπαίδευσης και ελέγχου ανά εποχή για το μοντέλο VGG16.



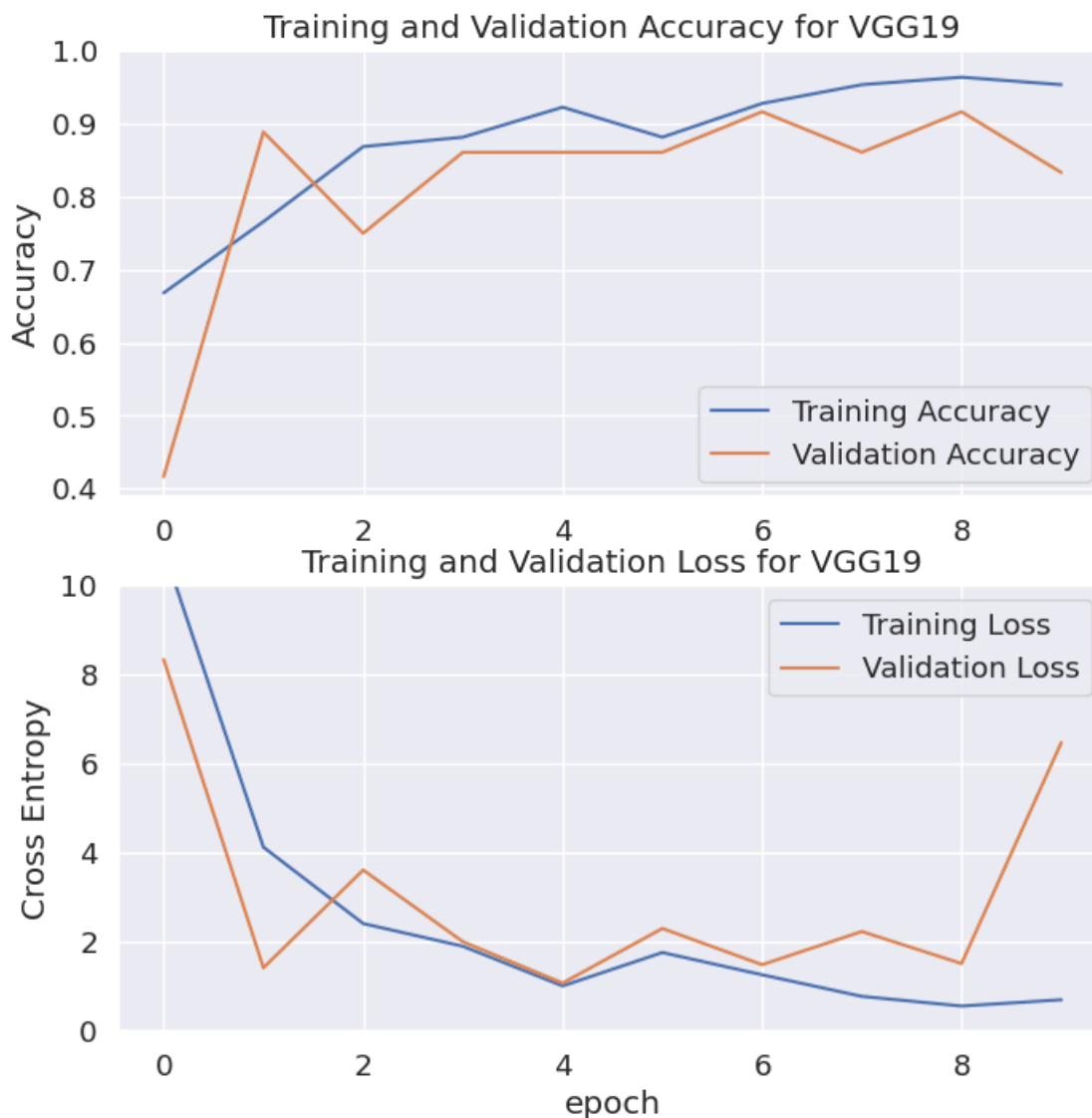
Εικόνα 3.27. β) Γραφική αναπαράσταση της ακρίβειας και του σφάλματος για τα δεδομένα εκπαίδευσης και ελέγχου ανά εποχή για το μοντέλο ResNet50.



Εικόνα 3.27. γ) Γραφική αναπαράσταση της ακρίβειας και του σφάλματος για τα δεδομένα εκπαίδευσης και ελέγχου ανά εποχή για το μοντέλο InceptionV3.



Εικόνα 3.27. δ) Γραφική αναπαράσταση της ακρίβειας και του σφάλματος για τα δεδομένα εκπαίδευσης και ελέγχου ανά εποχή για το μοντέλο Efficientnetb0.



Εικόνα 3.27. ε) Γραφική αναπαράσταση της ακρίβειας και του σφάλματος για τα δεδομένα εκπαίδευσης και ελέγχου ανά εποχή για το μοντέλο VGG19.

3.7.1 Fine Tuning

Χρησιμοποιούμε fine tuning με τα δεδομένα για να αυξήσουμε την επίδοση των μοντέλων μας. Ομοίως με πριν ξεπαγώνουμε ένα τμήμα του βασικού μοντέλου για να εκπαιδευτούν στα δεδομένα μας.

```
print("Number of layers in the base model: ", len(base_model1.layers))

# Fine-tune from this layer onwards
fine_tune_at = 16

# Freeze all the layers before the `fine_tune_at` layer
for layer in base_model1.layers[:fine_tune_at]:
    layer.trainable = False

Number of layers in the base model: 19
```

α) vgg16.

```
print("Number of layers in the base model: ", len(base_model2.layers))

# Fine-tune from this layer onwards
fine_tune_at = 169

# Freeze all the layers before the `fine_tune_at` layer
for layer in base_model2.layers[:fine_tune_at]:
    layer.trainable = False

Number of layers in the base model: 175
```

β) resnet50.

```
print("Number of layers in the base model: ", len(base_model3.layers))

# Fine-tune from this layer onwards
fine_tune_at = 294

# Freeze all the layers before the `fine_tune_at` layer
for layer in base_model3.layers[:fine_tune_at]:
    layer.trainable = False

Number of layers in the base model: 311
```

γ) Inceptionv3.

```
print("Number of layers in the base model: ", len(base_model4.layers))

# Fine-tune from this layer onwards
fine_tune_at = 230

# Freeze all the layers before the `fine_tune_at` layer
for layer in base_model4.layers[:fine_tune_at]:
    layer.trainable = False

Number of layers in the base model: 238
```

δ) EfficientNetB0.


```
print("Number of layers in the base model: ", len(base_model5.layers))

# Fine-tune from this layer onwards
fine_tune_at = 17

# Freeze all the layers before the `fine_tune_at` layer
for layer in base_model5.layers[:fine_tune_at]:
    layer.trainable = False

Number of layers in the base model: 22
```

ε) VGG19.

Παρατηρούμε ότι οι μεταβλητές που μπορούν να εκπαιδευτούν έχουν αυξηθεί από 6 σε 10, 12, 14, 16, 14 για το vgg16, resnet50, inceptionv3, efficientnetb0, vgg19 αντίστοιχα και έχουμε αύξηση στην ακρίβεια στα δεδομένα εκπαίδευσης και στα δεδομένα ελέγχου.

```
len(model1.trainable_variables)
10

len(model2.trainable_variables)
12

len(model3.trainable_variables)
14

len(model4.trainable_variables)
16

len(model5.trainable_variables)
14
```

Το callback σταμάτησε πρόωρα την εκπαίδευση στο μοντέλο με vgg16 .

```
39/39 [=====] - 8s 163ms/step - loss: 0.0834 - accuracy: 0.9949 - val_loss: 2.0653 - val_accuracy: 0.9167
Epoch 15/20
39/39 [=====] - ETA: 0s - loss: 1.6308e-04 - accuracy: 1.0000
Reached 99.5% accuracy so cancelling training!
39/39 [=====] - 9s 180ms/step - loss: 1.6308e-04 - accuracy: 1.0000 - val_loss: 0.9765 - val_accuracy: 0.9722
```

α) vgg16

```
Epoch 19/20
39/39 [=====] - 9s 202ms/step - loss: 0.0671 - accuracy: 0.9949 - val_loss: 1.3491 - val_accuracy: 0.8889
Epoch 20/20
39/39 [=====] - 9s 194ms/step - loss: 0.1725 - accuracy: 0.9923 - val_loss: 0.9012 - val_accuracy: 0.9167
```

β) resnet50.

```
Epoch 19/20
39/39 [=====] - 7s 159ms/step - loss: 0.0914 - accuracy: 0.9820 - val_loss: 5.5343 - val_accuracy: 0.6667
Epoch 20/20
39/39 [=====] - 9s 191ms/step - loss: 0.0706 - accuracy: 0.9897 - val_loss: 4.8630 - val_accuracy: 0.6389
```

γ) inceptionv3.

```
Epoch 19/20
39/39 [=====] - 8s 177ms/step - loss: 0.3304 - accuracy: 0.9794 - val_loss: 1.1107 - val_accuracy: 0.8611
Epoch 20/20
39/39 [=====] - 7s 158ms/step - loss: 0.2105 - accuracy: 0.9589 - val_loss: 0.9450 - val_accuracy: 0.8889
```

δ) efficientnetb0.

```
Epoch 19/20
39/39 [=====] - 10s 215ms/step - loss: 0.0270 - accuracy: 0.9923 - val_loss: 2.5399 - val_accuracy: 0.8889
Epoch 20/20
39/39 [=====] - ETA: 0s - loss: 0.0852 - accuracy: 0.9974
Reached 99.5% accuracy so cancelling training!
39/39 [=====] - 8s 172ms/step - loss: 0.0852 - accuracy: 0.9974 - val_loss: 1.4978 - val_accuracy: 0.8611
```

ε) vgg19.



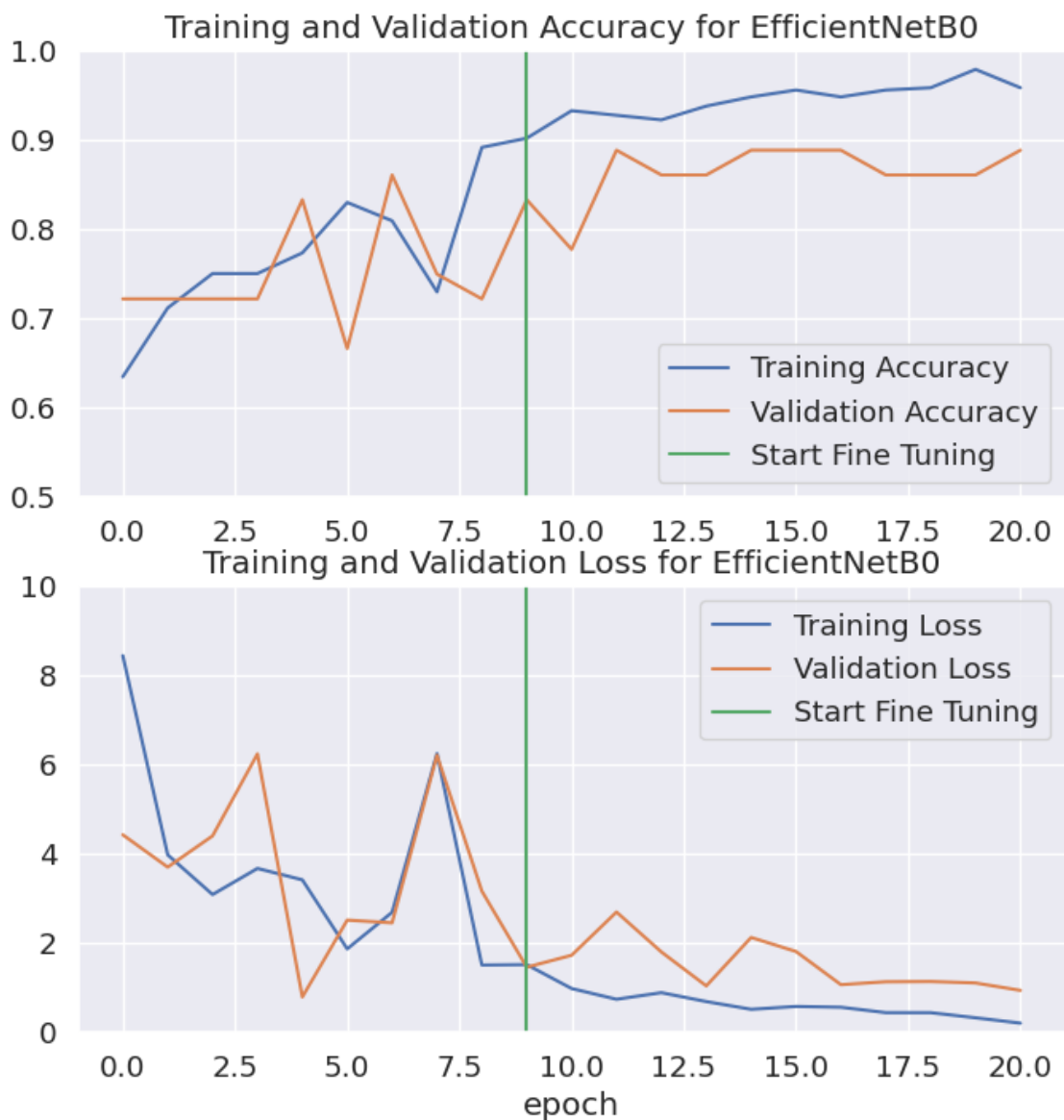
Εικόνα 3.28. α) Γραφική αναπαράσταση της ακρίβειας και του σφάλματος για τα δεδομένα εκπαίδευσης και ελέγχου ανά εποχή πριν και μετά το fine tuning για το μοντέλο VGG16.



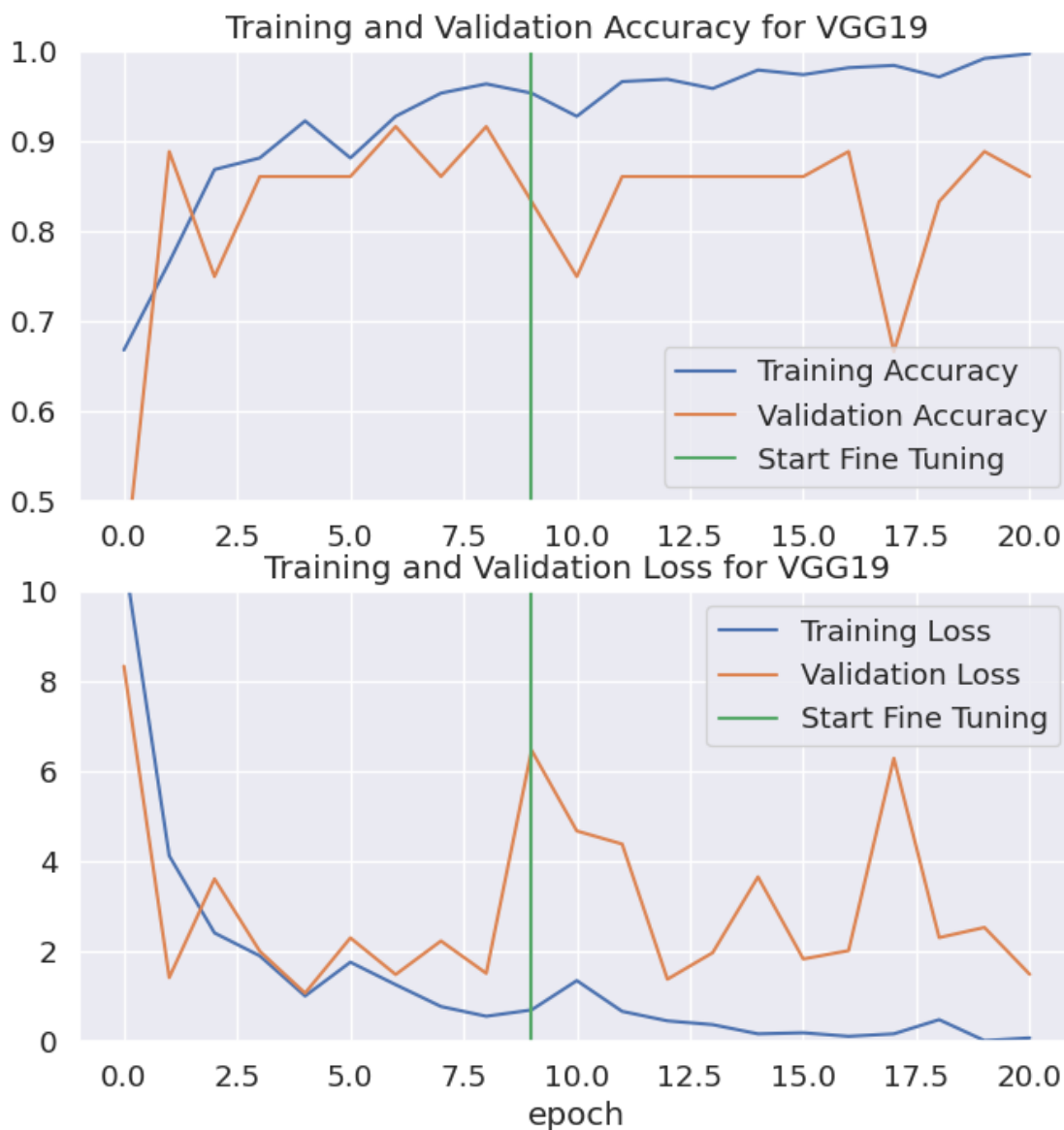
Εικόνα 3.28. β) Γραφική αναπαράσταση της ακρίβειας και του σφάλματος για τα δεδομένα εκπαίδευσης και ελέγχου ανά εποχή πριν και μετά το fine tuning για το μοντέλο ResNet50.



Εικόνα 3.28. γ) Γραφική αναπαράσταση της ακρίβειας και του σφάλματος για τα δεδομένα εκπαίδευσης και ελέγχου ανά εποχή πριν και μετά το fine tuning για το μοντέλο Inception V3.



Εικόνα 3.28. δ) Γραφική αναπαράσταση της ακρίβειας και του σφάλματος για τα δεδομένα εκπαίδευσης και ελέγχου ανά εποχή πριν και μετά το fine tuning για το μοντέλο EfficientNetb0.



Εικόνα 3.28. ε) Γραφική αναπαράσταση της ακρίβειας και του σφάλματος για τα δεδομένα εκπαίδευσης και ελέγχου ανά εποχή πριν και μετά το fine tuning για το μοντέλο VGG19.

3.7.2 - Εξαγωγή συμπερασμάτων

Παρατηρούμε ότι μεγαλύτερη ακρίβεια στα δεδομένα ελέγχου την έχουμε με το μοντέλο το οποίο έχει κατασκευαστεί από το vgg16 βασικό μοντέλο.

```
loss, accuracy = model1.evaluate(test_dataset)
print('Test accuracy :', accuracy)

4/4 [=====] - 1s 73ms/step - loss: 0.9765 - accuracy: 0.9722
Test accuracy : 0.97222220889767456

loss, accuracy = model2.evaluate(test_dataset)
print('Test accuracy :', accuracy)

4/4 [=====] - 1s 38ms/step - loss: 0.9012 - accuracy: 0.9167
Test accuracy : 0.9166666865348816

loss, accuracy = model3.evaluate(test_dataset)
print('Test accuracy :', accuracy)

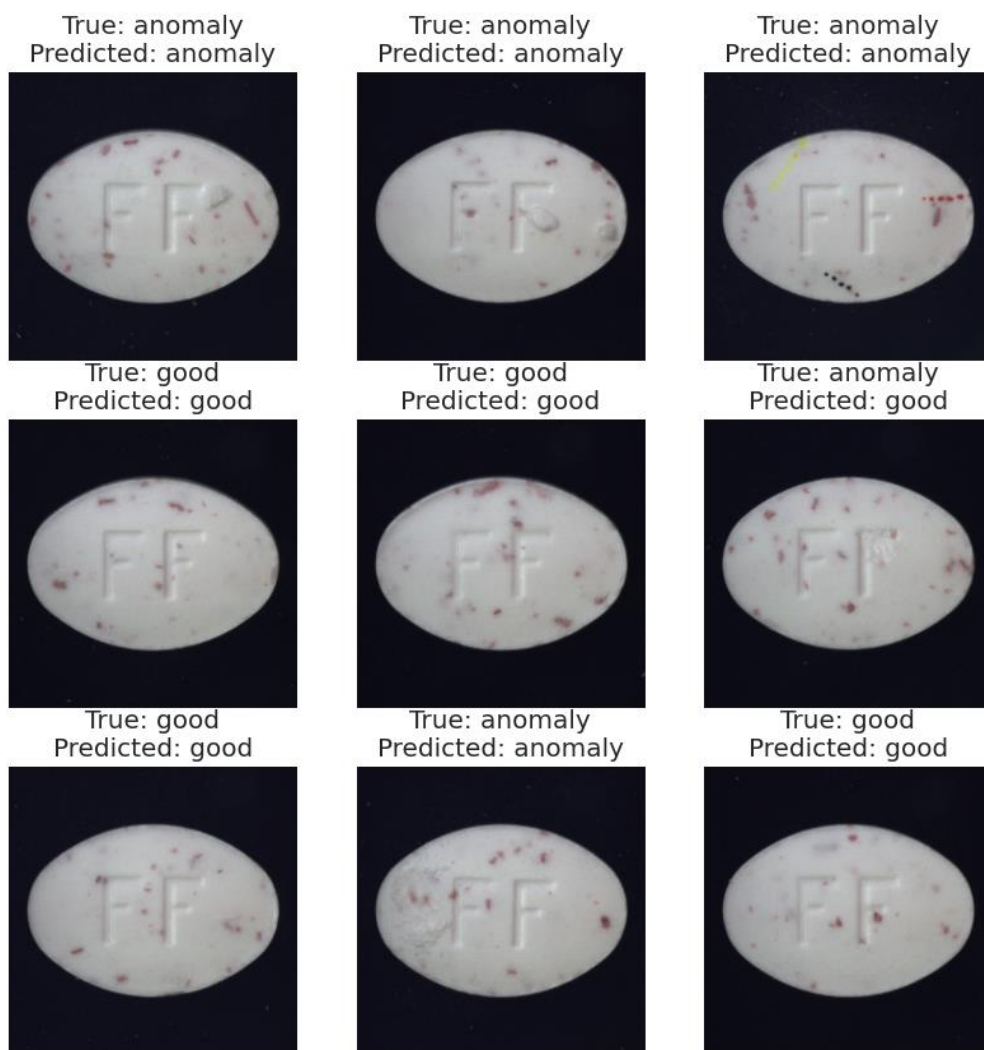
4/4 [=====] - 1s 32ms/step - loss: 4.8635 - accuracy: 0.6389
Test accuracy : 0.6388888955116272

loss, accuracy = model4.evaluate(test_dataset)
print('Test accuracy :', accuracy)

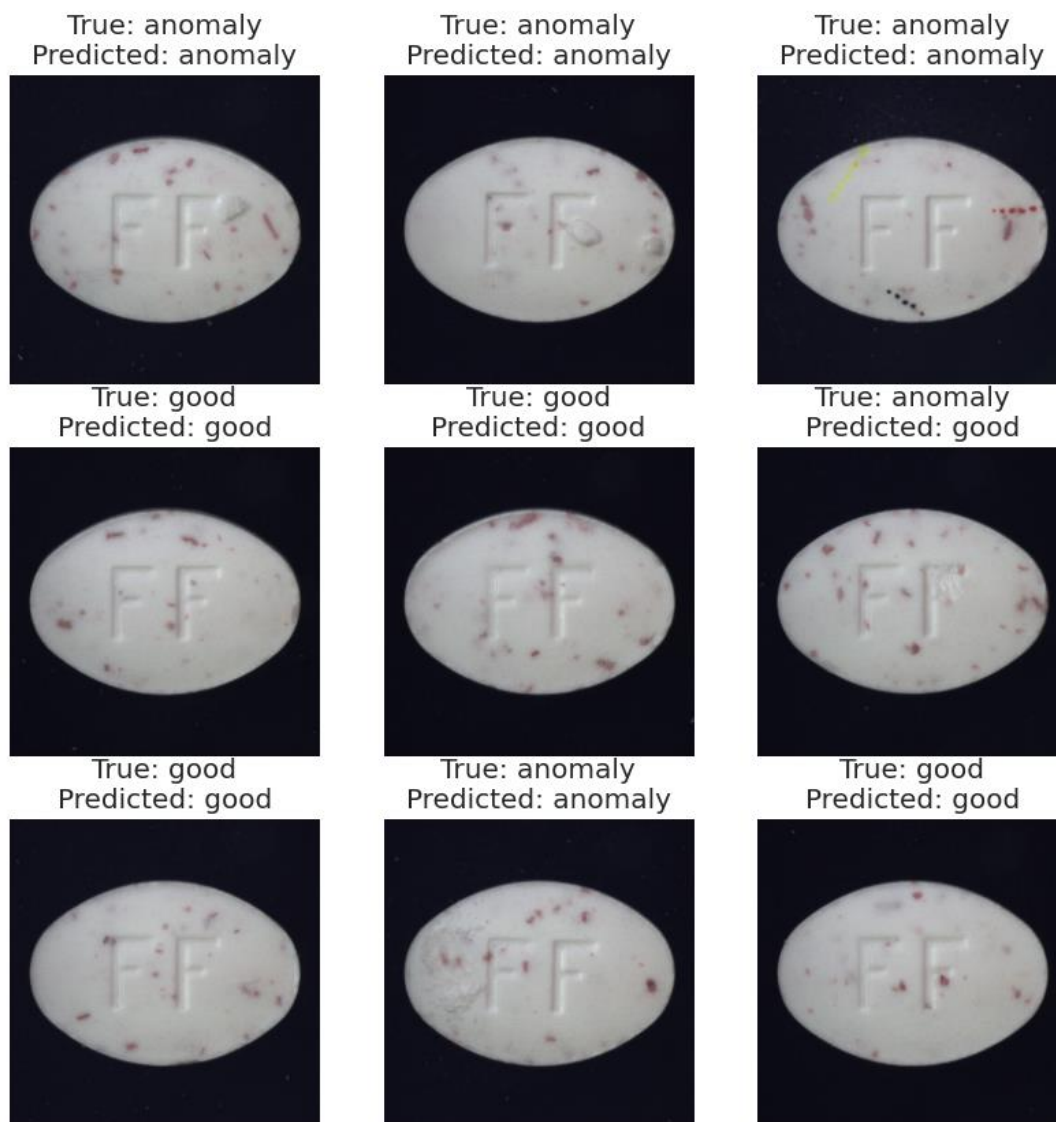
4/4 [=====] - 1s 29ms/step - loss: 0.9450 - accuracy: 0.8889
Test accuracy : 0.8888888955116272

loss, accuracy = model5.evaluate(test_dataset)
print('Test accuracy :', accuracy)

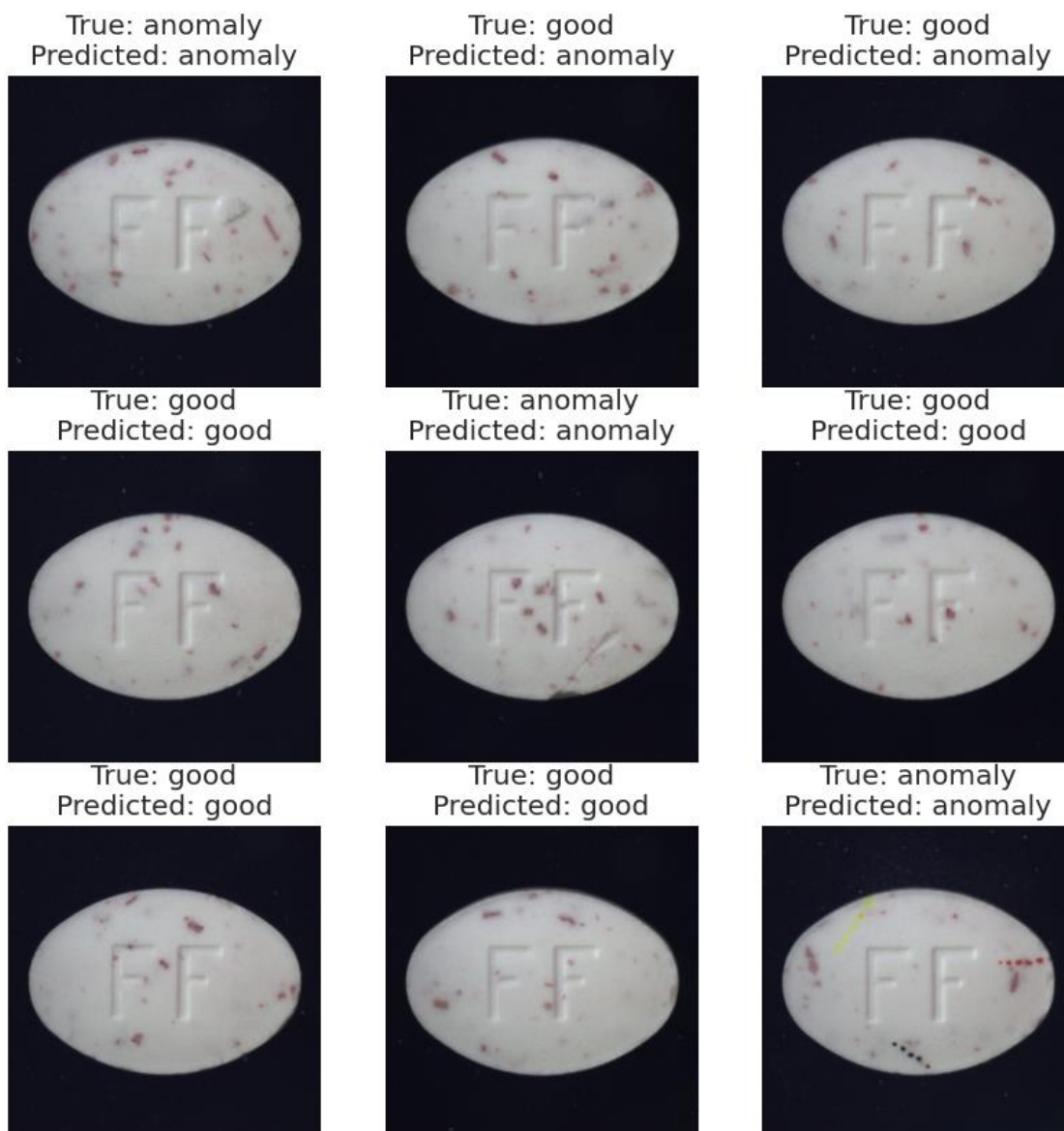
4/4 [=====] - 1s 74ms/step - loss: 1.4978 - accuracy: 0.8611
Test accuracy : 0.8611111044883728
```

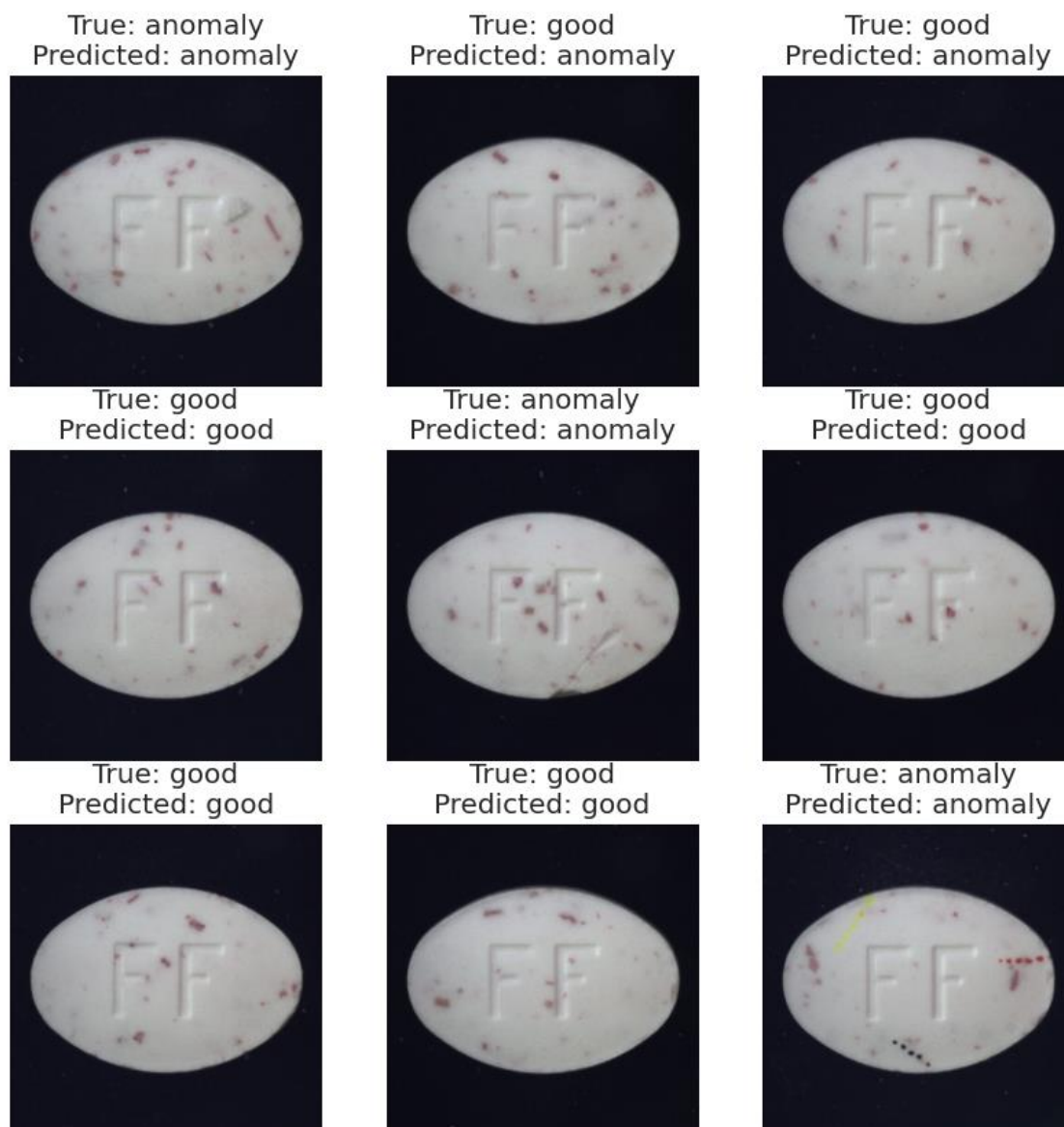
Εικόνα 3.29. α) Γραφική αναπαράσταση προβλεπομένων και πραγματικών κατηγοριών για 9 εικόνες από δεδομένα ελέγχου για vgg16.



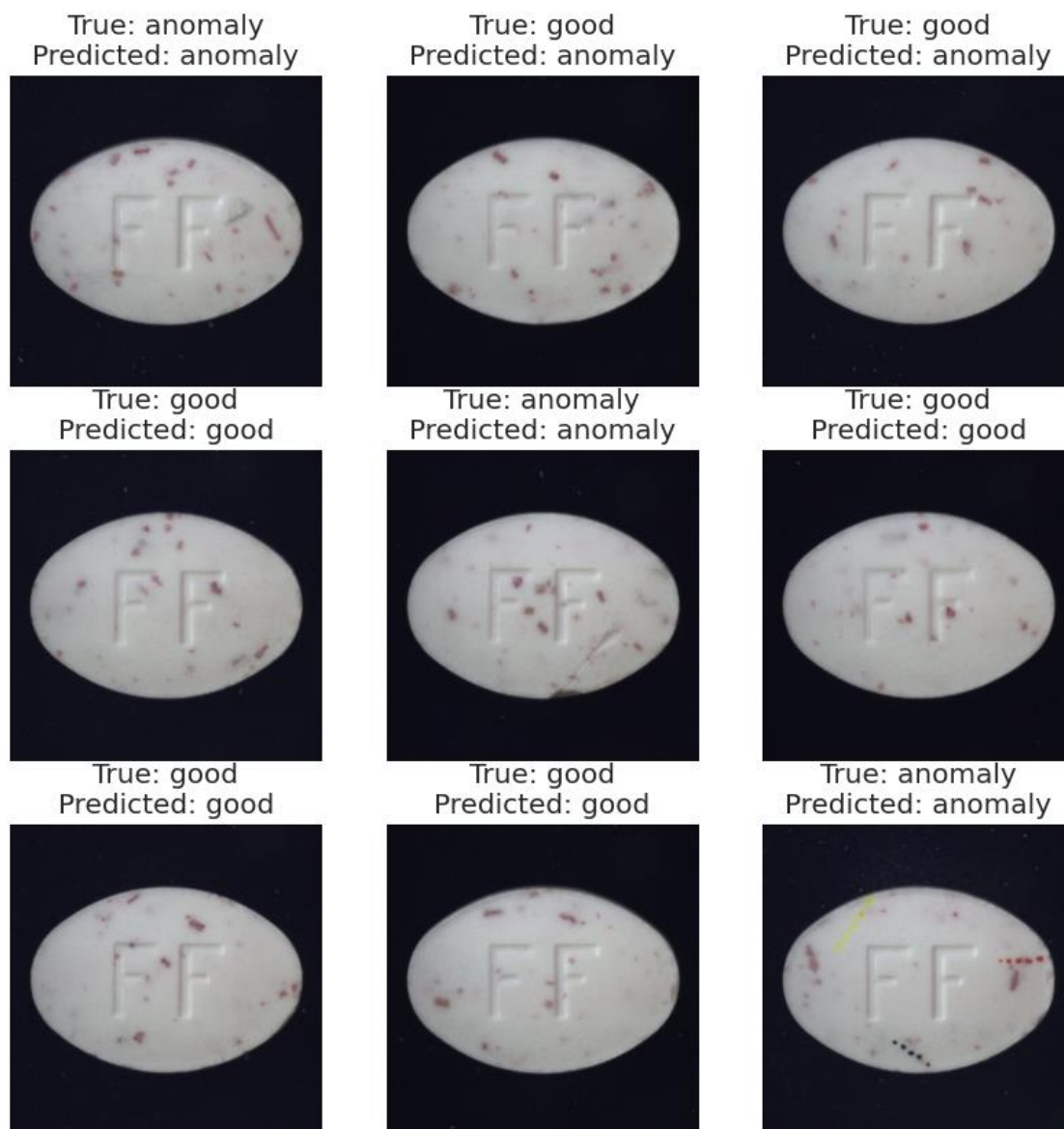
Εικόνα 3.29. β) Γραφική αναπαράσταση προβλεπομένων και πραγματικών κατηγοριών για 9 εικόνες από δεδομένα ελέγχου για resnet50.



Εικόνα 3.29. γ) Γραφική αναπαράσταση προβλεπομένων και πραγματικών κατηγοριών για 9 εικόνες από δεδομένα ελέγχου για inception v3.



Εικόνα 3.29. δ) Γραφική αναπαράσταση προβλεπομένων και πραγματικών κατηγοριών για 9 εικόνες από δεδομένα ελέγχου για efficientnetb0.



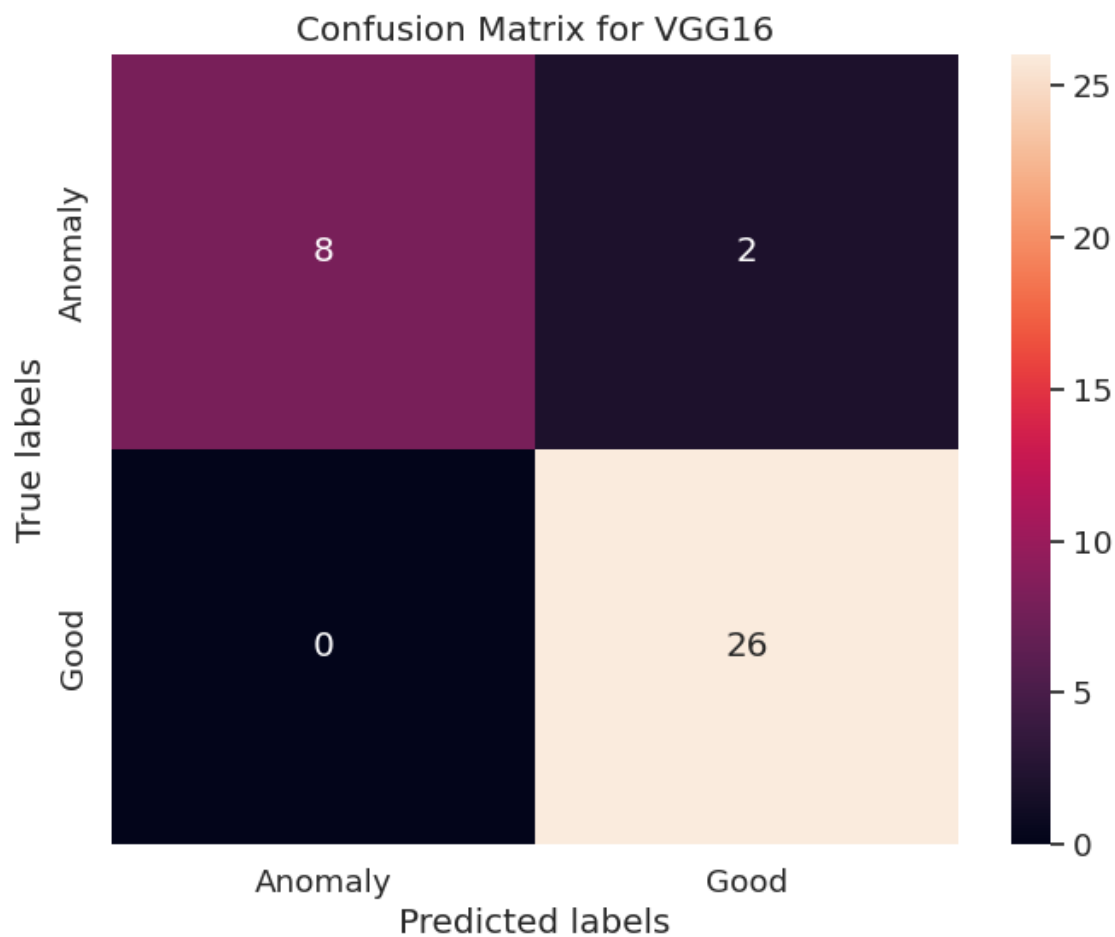
Εικόνα 3.29. ε) Γραφική αναπαράσταση προβλεπομένων και πραγματικών κατηγοριών για 9 εικόνες από δεδομένα ελέγχου για vgg19.

3.7.3 - Στατιστικά στοιχεία

	Precision	Recall	F1-score	Support
Anomaly	1.000	0.800	0.889	10
Good	0.929	1.000	0.963	26

Accuracy			0.972	36
Macro Avg	0.964	0.900	0.926	36
Weighted Avg	0.948	0.944	0.942	36

Πίνακας 3.2. α) Στατιστικά στοιχεία ταξινόμησης για vgg16.

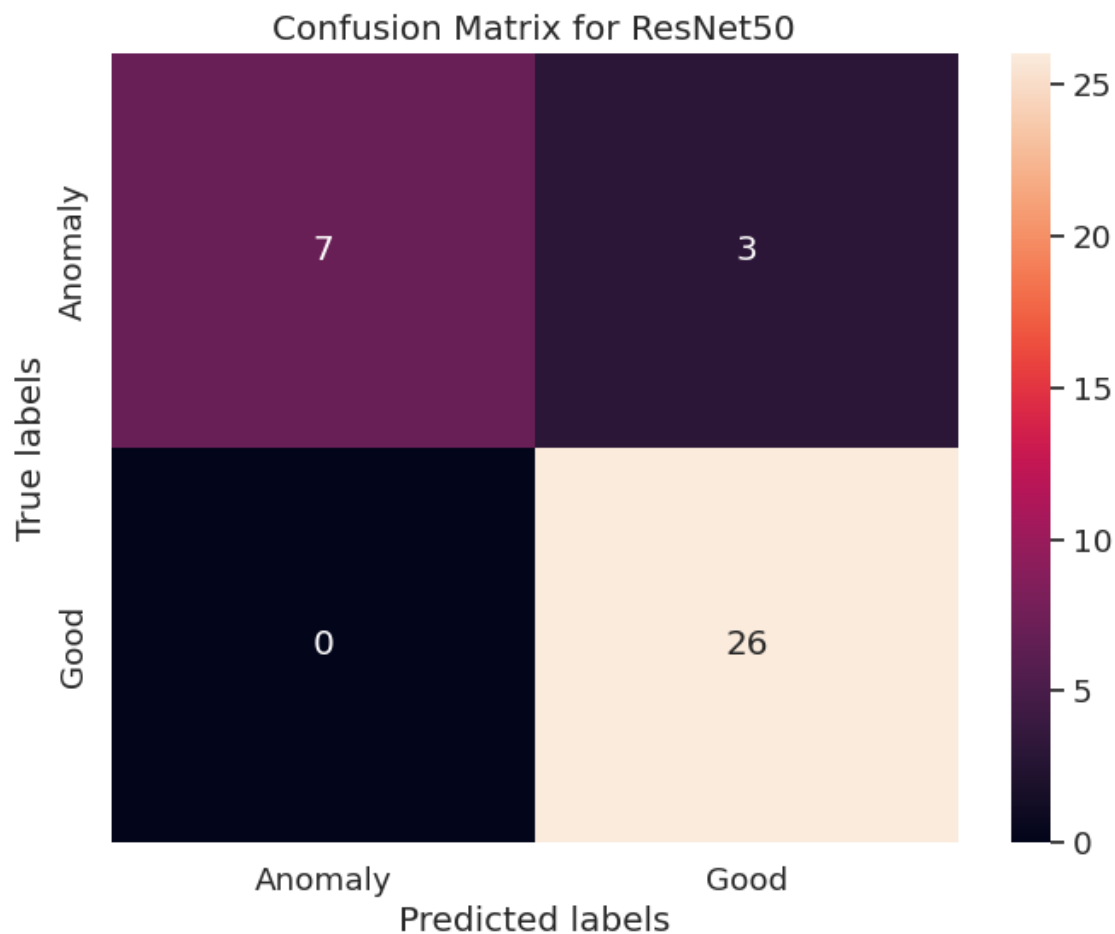


Εικόνα 3.30. α) Μήτρα σύγκρισης για το μοντέλο vgg16.

	Precision	Recall	F1-score	Support
Anomaly	1.000	0.700	0.824	10
Good	0.897	1.000	0.945	26
Accuracy			0.916	36

Macro Avg	0.948	0.850	0.884	36
Weighted Avg	0.925	0.917	0.912	36

Πίνακας 3.2. β) Στατιστικά στοιχεία ταξινόμησης για resnet50.

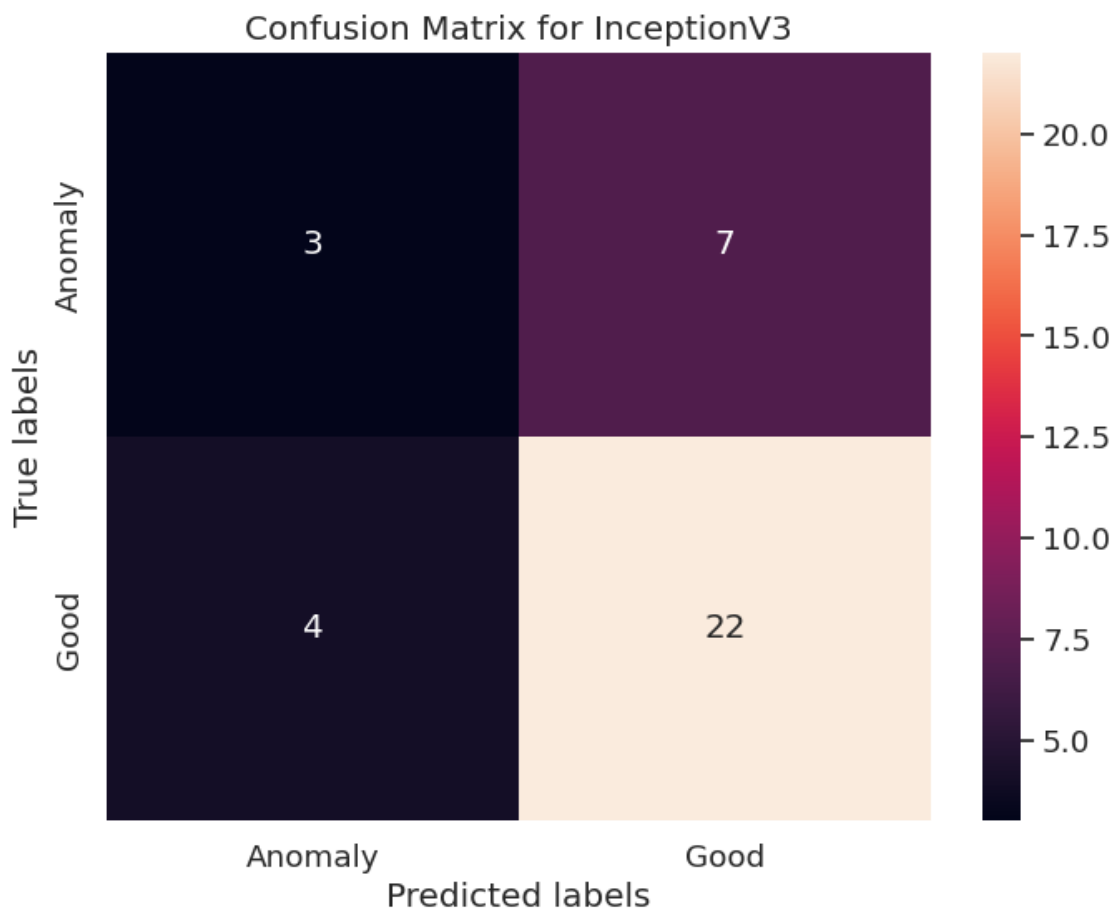


Εικόνα 3.30. β) Μήτρα σύγκρισης για το ResNet50.

	Precision	Recall	F1-score	Support
Anomaly	0.429	0.300	0.353	10
Good	0.756	0.846	0.800	26
Accuracy			0.638	36
Macro Avg	0.594	0.573	0.576	36

Weighted Avg	0.664	0.694	0.676	36
---------------------	-------	-------	-------	----

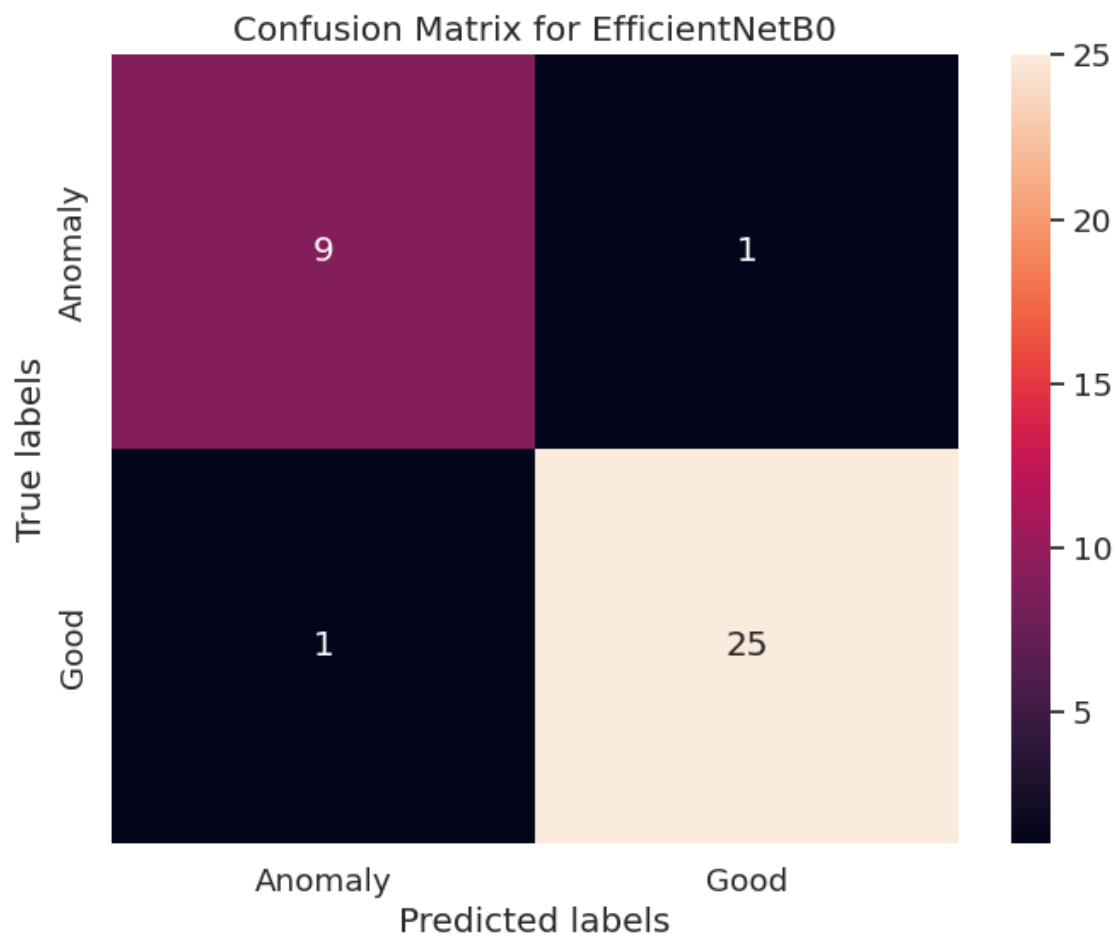
Πίνακας 3.2. γ) Στατιστικά στοιχεία ταξινόμησης για inception v3.



Εικόνα 3.30. γ) Μήτρα σύγχυσης για το Inception V3.

	Precision	Recall	F1-score	Support
Anomaly	0.900	0.900	0.900	10
Good	0.962	0.962	0.962	26
Accuracy			0.888	36
Macro Avg	0.931	0.931	0.931	36
Weighted Avg	0.944	0.944	0.944	36

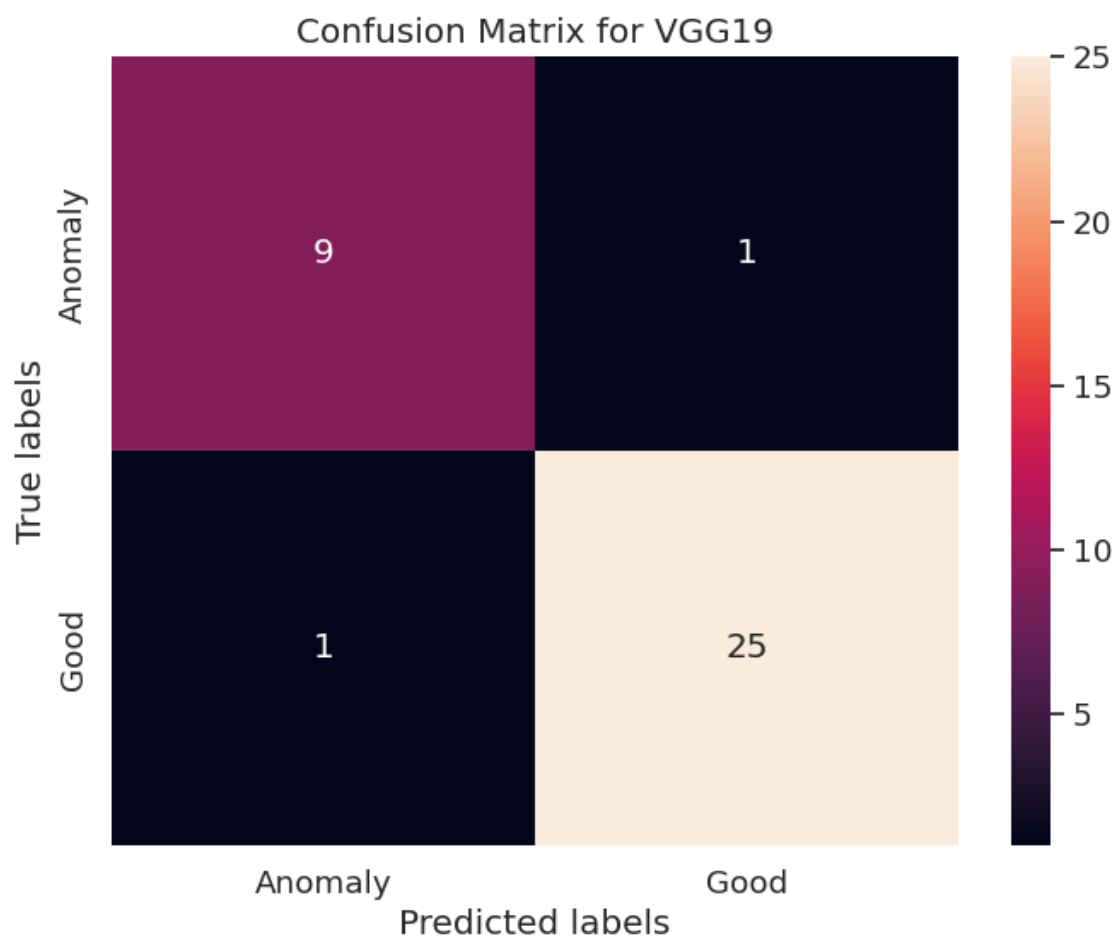
Πίνακας 3.2. δ) Στατιστικά στοιχεία ταξινόμησης για efficientnetb0.



Εικόνα 3.30. δ) Μήτρα σύγκρισης για το Efficientnetb0.

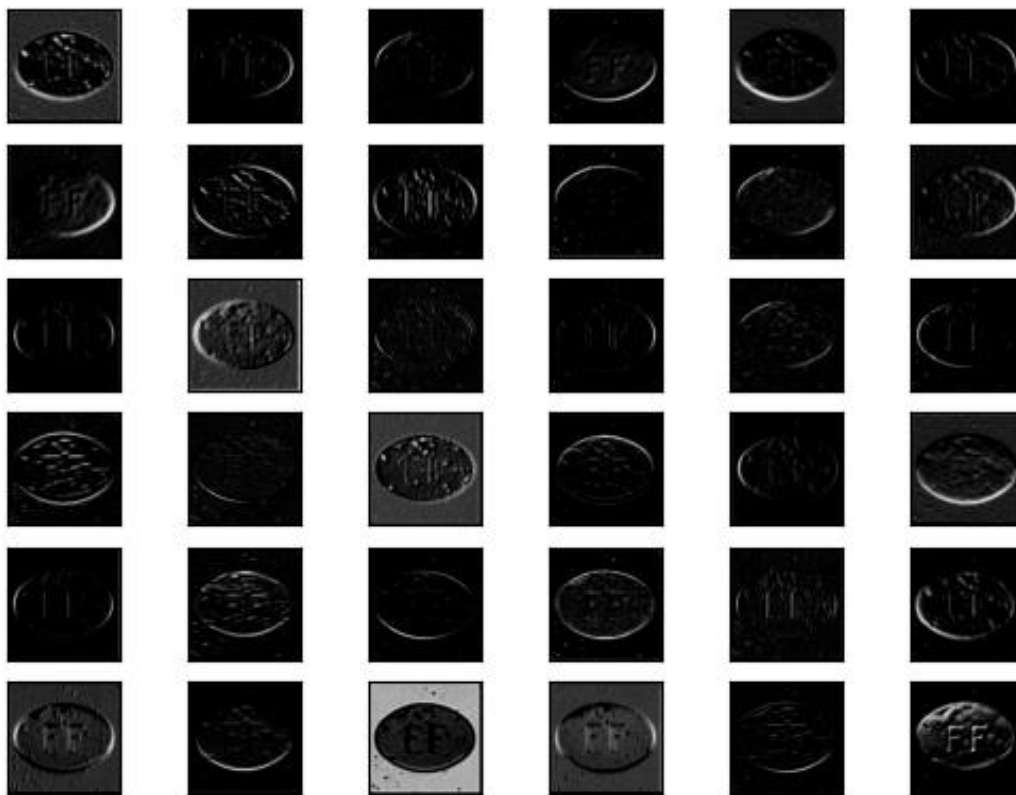
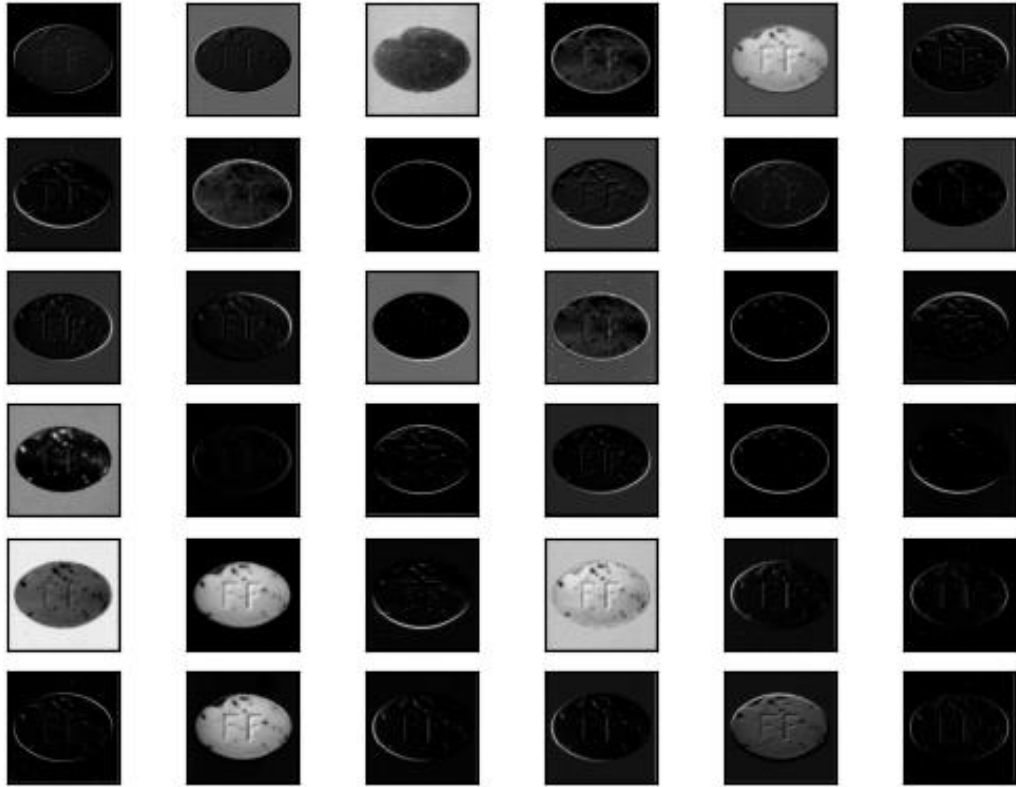
	Precision	Recall	F1-score	Support
Anomaly	0.898	0.898	0.898	10
Good	0.960	0.960	0.960	26
Accuracy			0.861	36
Macro Avg	0.930	0.930	0.930	36
Weighted Avg	0.940	0.940	0.940	36

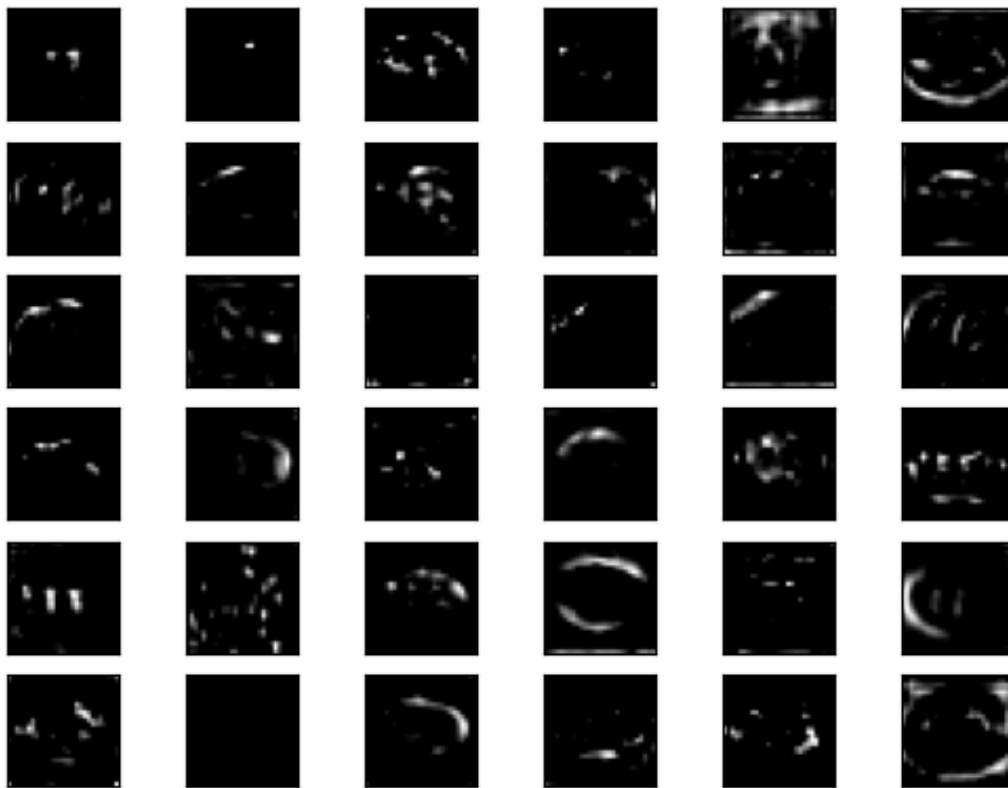
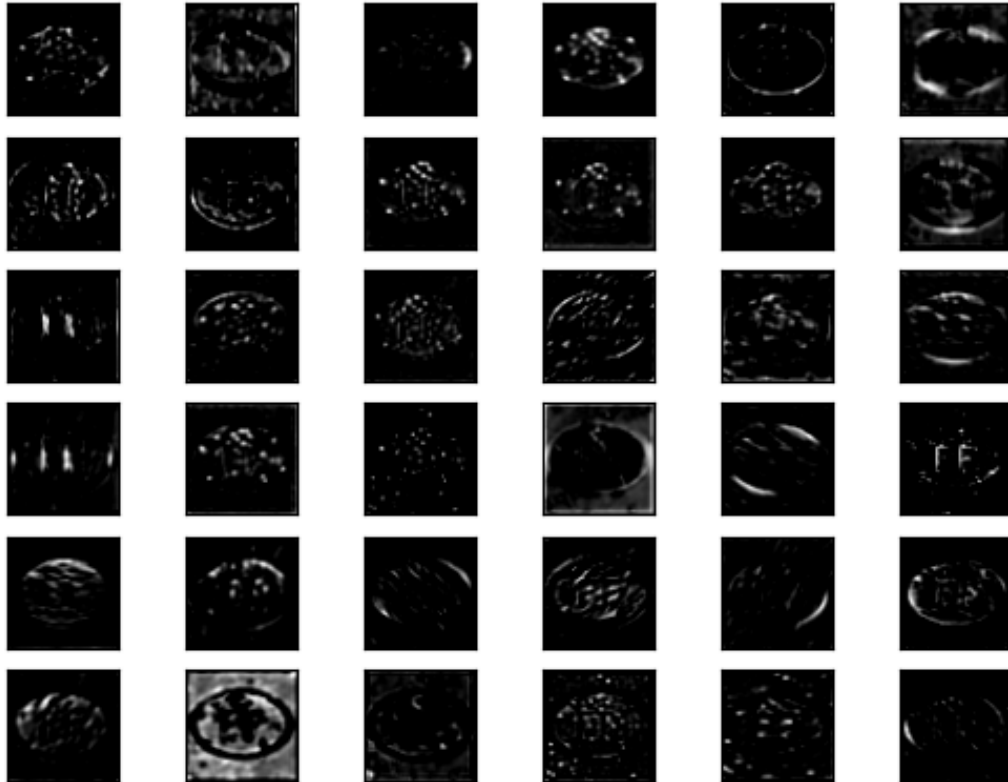
Πίνακας 3.2. ε) Στατιστικά στοιχεία ταξινόμησης για vgg19.

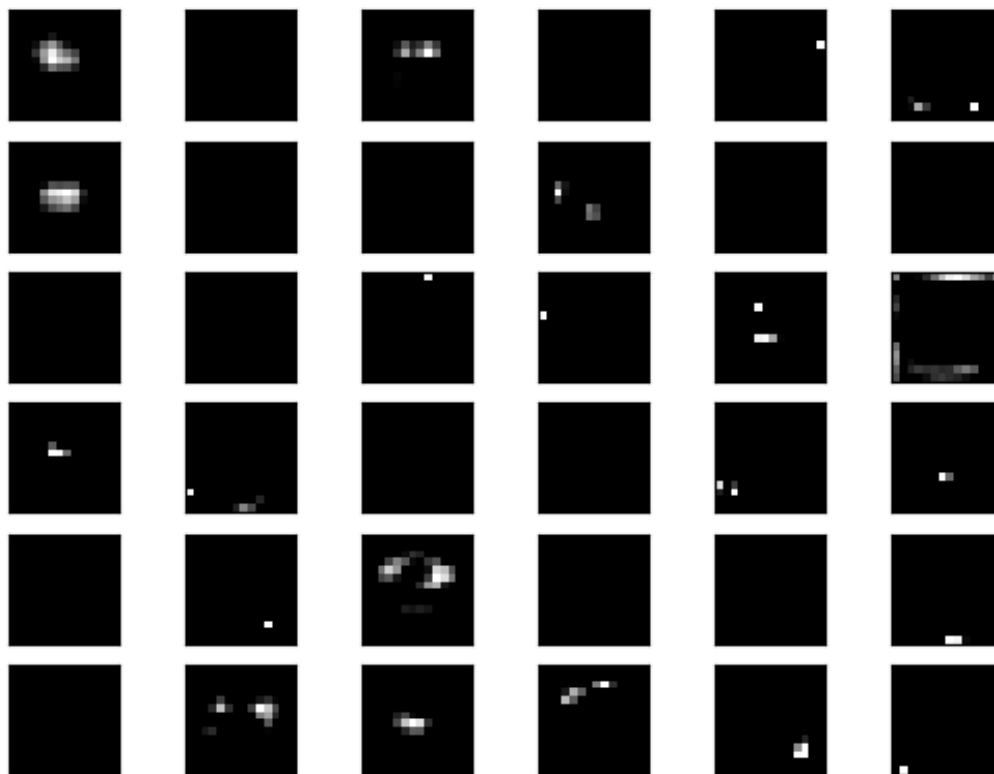


Εικόνα 3.30. ε) Μήτρα σύγχυσης για το VGG19.

Απεικόνιση ενός δείγματος από τους χάρτες χαρακτηριστικών για μια εικόνα είσοδος για το μοντέλο vgg16 που εκπαιδεύτηκε.







Εικόνα 3.31. α) Block 1- convolution 2. β) Block 2- convolution 2. γ) Block 3- convolution 3. δ) Block 4- convolution 3. ε) Block 5- convolution 3. [28]

3.8 - Εξαγωγή μοντέλων

Τελικός εκπαιδεύουμε το καλύτερο μοντέλο μας με το `test_dataset`, έπειτα αποθηκεύουμε και μετασχηματίζουμε το μοντέλο σε `onnx format` το οποίο μας δίνει την ικανότητα να εφαρμόσουμε το μοντέλο σε μια μεγάλη ποικιλία από υπολογιστικές πλατφόρμες βαθιάς μάθησης όπως του CoreML για iOS συσκευές.

```
#train teh best model with the test set
history_final = model1.fit(test_dataset, epochs=10)

import tf2onnx
#save the model
model1.save("vgg16_custom")
# Load the TensorFlow SavedModel
tf_model_path = "/content/vgg16_custom"
onnx_model_path = "/content/.onnx"

# Convert the TensorFlow model to ONNX format
onnx_model, _ = tf2onnx.convert.from_saved_model(tf_model_path)

# Save the ONNX model to a file
with open(onnx_model_path, "wb") as f:
    f.write(onnx_model.SerializeToString())

print(f"Model successfully converted to ONNX and saved to {onnx_model_path}")
```

Είναι σημαντικό τα μοντέλα που κατασκευάζονται να μην κρίνονται μόνο ως προς την επίδοση τους αλλά και ως προς την ταχύτητα κατηγοριοποίησης που μπορούν να επιτύχουν όταν εφαρμοστούν. Συγκεκριμένα από τα μοντέλα που εξετάστηκαν τα μεγαλύτερο σε μέγεθος μνήμης είναι τα vgg16, vgg19 επίσης έχουν και τις χαμηλότερες ταχύτητες κατηγοριοποίησης σε cpu καθιστώντας τα μη ιδανικούς αντιπροσώπους σε συσκευές που δεν μπορούν να αξιοποιήσουν gpu. Αντιθέτως τα MobileNetv2 και EfficientNetB0 έχουν όχι μόνο το μικρότερο μέγεθος μνήμης, από τα μοντέλα που εξετάστηκαν αλλά και τις πιο γρήγορες ταχύτητες κατηγοριοποίησης σε cpu. Το top-1 και top-5 ακρίβεια αναφέρεται στην ακρίβεια του μοντέλου στο imagenet σετ δεδομένων ελέγχου. Το βάθος αναφέρεται στον αριθμό των στρωμάτων με παραμέτρους.

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time(ms) per inference step(CPU)	Time(ms) per inference step(GPU)
VGG16	528	71.3%	90.1%	138.4M	16	69.5	4.2
VGG19	549	71.3%	90.0%	143.7M	19	84.8	4.4
ResNet50	98	74.9%	92.1%	25.6M	107	58.2	4.6

InceptionV3	92	77.9%	93.7%	23.9M	189	42.2	6.9
MobileNetV2	14	71.3%	90.1%	3.5M	105	25.9	3.8
EfficientNetB0	29	77.1%	93.3%	5.3M	132	46.0	4.9

Πίνακας 3.3. Στατιστικά των μοντέλων που χρησιμοποιήθηκαν σαν βάση για την εφαρμογή [37].

Συμπεράσματα Εφαρμογής

Συμπερασματικά, η παρούσα εφαρμογή έχει αναλάβει μια ολοκληρωμένη διερεύνηση της ταξινόμησης ελαττωματικών χαπιών με την χρήση της βαθιάς μάθησης στη φαρμακευτική παραγωγή, εστιάζοντας ιδιαίτερα στον ρόλο της στη βελτίωση των διαδικασιών ποιοτικού ελέγχου για την παραγωγή δισκίων.

Χρησιμοποιήθηκαν μια μεγάλη ποικιλία από τα πιο δημοφιλή μοντέλα υπολογιστικής όρασης για την ταξινόμηση εικόνων και βρέθηκε ότι η αρχιτεκτονική βασισμένη στο vgg16 είχε την καλύτερη επίδοση. Παρόλα αυτά πρέπει να αναφερθεί ότι τα υπόλοιπα μοντέλα είχαν πολύ καλή επίδοση με άξιο αναφοράς το EfficientNetb0 και το MobileNetv2 τα οποία είναι πολύ μικρότερα στο μέγεθος τους σε σχέση με το vgg16.

Είναι σημαντικό να αναφερθούν όμως οι δυσκολίες που αντιμετωπίστηκαν κατά την διάρκεια της πρακτικής εφαρμογής. Το μεγαλύτερο εμπόδιο ήταν στην εύρεση κατάλληλων δεδομένων για την επιτυχή εκπαίδευση του μοντέλου. Συγκεκριμένα οι εικόνες από τα πιθανά σφάλματα φαρμακευτικών προϊόντων είναι συνήθως μη διαθέσιμες στο κοινό αφού μπορεί να εμπεριέχουν εταιρικά μυστικά και σημαντικές πληροφορίες για την αντιστοιχεί γραμμή παραγωγής. Επίσης από τις ανοιχτά διαθέσιμες βάσης δεδομένων ελάχιστες έχουν ενδείξεις από ειδήμονες στο αντίστοιχο πεδίο και είναι αρκετά μεγάλες σε μέγεθος για να εκπαιδεύσουν ένα βαθύ νευρωνικό δίκτυο. Αυτό αντιμετωπίστηκε σε μεγάλο βαθμό από την προεπεξεργασία των εικόνων (image augmentation).

Οι γνώσεις που προέκυψαν από αυτήν την έρευνα δείχνουν ότι οι τεχνολογίες υπολογιστικής όρασης μπορούν προσφέρουν ένα αξιόπιστο, ευέλικτο, φθινό και αποτελεσματικό μέσο για τον έλεγχο της ποιότητας των φαρμακευτικών παρασκευασμάτων αν δοθούν μια ικανοποιητική ποικιλία από δεδομένα εκπαίδευσης, για τον εντοπισμό ελαττωμάτων και την τήρηση αυστηρών προτύπων παραγωγής.

Συμπεράσματα




Συμπερασματικά, η παρούσα διατριβή έχει αναλύσει στην πολύπλευρη σφαίρα της βαθιάς μάθησης και τον μετασχηματιστικό αντίκτυπό της σε διάφορες βιομηχανικές εφαρμογές. Μέσα από μια ολοκληρωμένη διερεύνηση διαφορετικών τομέων, έχει κατέστη προφανές ότι η ικανότητα της βαθιάς μάθησης να αναλύει πολύπλοκα σύνολα δεδομένων, να εξάγει περίπλοκα μοτίβα και να κάνει ενημερωμένες προβλέψεις θα επιφέρει μια νέα εποχή αποτελεσματικότητας, ακρίβειας και καινοτομίας. Από τη βελτίωση των διαδικασιών παραγωγής και τον ποιοτικό έλεγχο μέχρι την στην προγνωστική συντήρηση, τη βελτιστοποίηση της εφοδιαστικής αλυσίδας και τη ρομποτική, η βαθιά μάθηση έχει αποδείξει τις αξιοσημείωτες δυνατότητές της να αναμορφώσει τις βιομηχανίες.





Μέσω της ενσωμάτωσης των τεχνικών υπολογιστικής όρασης, οι κατασκευαστές φαρμακευτικών προϊόντων έχουν την ικανότητα να αναλύουν σχολαστικά μεγάλους όγκους δεδομένων φαρμακευτικών παρασκευασμάτων με ελάχιστη ανθρώπινη παρέμβαση οδηγώντας σε αυξημένη ακρίβεια και συνέπεια στον εντοπισμό ελαττωμάτων. Αυτό, με τη σειρά του, μεταφράζεται σε βελτιωμένη ποιότητα προϊόντων, ελαχιστοποίηση των απορριμμάτων και βελτιωμένη λειτουργική απόδοση.


Μελλοντικές εργασίες μπορούν να εγγραφούν στην επέκταση των δεδομένων που χρησιμοποιήθηκαν με την χρήση τεχνικών image generation και περαιτέρω ταξινόμηση των δεδομένων σε όλες τις πιθανές κατηγορίες σφαλμάτων.

Παράρτημα 1




TABLETING PROCESS



DEFECTS	PROCESS RELATED CAUSES	FORMULATION RELATED CAUSES
<p>Capping</p> 	<p>Mainly problems related to poorly finished dies or shape of punches, i.e. deep concavity or bevelled-edge of the punch. Capping can also happen if the punch remains below the face of die during ejection. Other reasons may be in the incorrect adjustment of sweep-off blade or too high turret speed (too short dwell time).</p>	<p>Large number of fines in the granulation or segregation during the pressing procedure. Low moisture content and insufficient amount or improper binding material.</p>
<p>Lamination</p> 	<p>Rapid decompression and too high pressure force. Too high turret speed (too short dwell time).</p>	<p>Large number of fines in the granulation or segregation during the pressing procedure. Low moisture content and insufficient amount or improper binding material. Rapid relaxation of the peripheral regions of the tablet due to lack of compactness of the material (on ejection from die).</p>
<p>Chipping</p> 	<p>Reasons relating to the shape and condition of the die, for example worn die groove at compression point or barrelled die, and punch. The latter represents a problem if the edges are turned inward or the concavity is too deep to compress powder blend.</p>	<p>Too wet material, insufficient amount of lubricant or inappropriate binder leading into sticking. Improper convexity of the tablet which results in chipping during coating, dedusting, handling etc. (procedures that follow the tableting).</p>

<p>Binding</p> 	<p>Poorly finished dies and rough dies due to abrasion and corrosion, undersized dies having too little clearance. Another reason can also be too much pressure in the tablet press.</p>	<p>Reasons mainly relating to material properties of granules, as for example too high moisture, too coarse and too hard granules for the lubricant to be effective, abrasive material of the granules causing cutting into dies or granular material too warm thus sticking to the die. Insufficient and improper lubricant are also important factors.</p>
<p>Picking</p> 	<p>Free rotation of upper or lower punch during ejection of a tablet and rough or scratched punch faces. When embossing or engraving letters the latter can be problematic in case of B, A, O, R, P, Q and G. It can happen that the dividing lines on a punch are too deep or applied pressure is not high enough (soft tablet).</p>	<p>Excessive moisture in granules, insufficient or improper lubrication and too high amount of binder.</p>
<p>Thickness</p> 	<p>Punch tooling is of inconsistent length (between batches). Uncleaned and unmaintained tablet press.</p>	<p>Granulation or powder blend is not adequately consistent in particle and distribution size (inconsistent filling of the die).</p>
<p>Dots/dirt</p> 	<p>Contaminated material, e.g. punch lubricant (inadequate viscosity or used in excess) or scraped material from the bottom of the feeder. Scorched material from within the die bore or from the punch tips. Improper set-up or adjustment of the feeder causing too much overfill which is being returned to the feeder resulting in additional fines.</p>	<p>Contaminated material/formulation during the processes prior tableting.</p>

<p>Shape</p> 	<p>Too high press speeds and/or too short pre-compression.</p>	<p>Tablets are either too soft or too damp.</p>
--	--	---

COATING PROCESS

DEFECTS	PROCESS RELATED CAUSES	FORMULATION RELATED CAUSES
<p>Cracking</p> 	<p>Improper coating process conditions that lead to core expansion and consequently cracking of coating (excessive heating effects, low pan speed, improper air temperature and spray rate etc.).</p>	<p>Poor elasticity or lack of mechanical strength of coating to withstand the core expansion.</p>
<p>Sticking/satellite</p> 	<p>Over wetting during the application of the coating. Spray rate too high, pan speed and process temperature too low, bad spray gun alignment.</p>	<p>Coating formulation has low glass transition temperature, causing it to become sticky. Too low tablet cap curvature.</p>
<p>Colour deviation</p> 	<p>Uniformity of coating distribution is inadequate (bad spray gun setup, inconsistent gun or bed distance, low pan speed etc.).</p>	<p>Coverage properties of the coating are insufficient, solid content of the suspension is too high, weight gain or batch quantity is too low.</p>

<p>Dark spots</p> 	<p>Long duration of the coating process which increases the chance of scuffing/abrasion of pan wall. Other reasons may also lay in low pan speed or underfilling which can cause the tablets to slide in the pan.</p>	<p>Coating formulation contains a high concentration of materials (for example pigments like titanium dioxide) which are harder than the stainless-steel pan walls, causing an abrasion of the inox pan wall, resulting in the dark spots in the coating.</p>
<p>Damaged print</p> 	<p>Incorrect alignment of the tablet with the punch, too much ink on the punch, improper ink viscosity and applying pressure, uneven flow of the ink, clogged ink jets etc.</p>	<p>Improper formulation of the ink blends with a combination of coating ingredients.</p>

[30][31][32][33] [34][35].

Βιβλιογραφία

- [1]. Predictive Maintenance Solutions for the Steel & Iron Industry.
<https://news.iac-intl.com/predictive-maintenance-solutions-for-steel-iron-industry>
- [2]. Understanding Predictive Maintenance in Manufacturing - Daniel Faggella, (September 14, 2020) <https://emerj.com/ai-sector-overviews/predictive-maintenance-manufacturing/>
- [3]. NARIMUNE (31 March 2023). 3D Inspection.
<https://narimune.com/page-15501-3d2d-inspection.html#>
- [4]. Cognex | Food and beverage packaging inspection.
<https://www.cognex.com/industries/food-and-beverage/packaging-inspection>
- [5]. Παράνομες οι κάμερες παρακολούθησης σε χώρους εργασίας, λέει η Αρχή Προστασίας Προσωπικών Δεδομένων (8 Απριλίου 2014).
<https://www.taxheaven.gr/news/18226/paranomes-oi-kameres-parakoloyohshs-se-xwroys-ergasias-leei-h-arxh-prostasias-proswpikwn-dedomenwn>
- [6]. Cognex | Drug delivery device final assembly verification.
<https://www.cognex.com/industries/pharmaceuticals-medical/medical-device-solutions/drug-delivery-device-inspection>
- [7]. Industrial Vision | Robot inspection systems. IVS combine robots and machine vision (2023).
<https://www.industrialvision.co.uk/about-us/case-studies/robot-inspection-systems-ivs-combine-robots-and-machine-vision>
- [8]. Manufacturing AI Use Cases and Trends – An Executive Brief, Daniel Faggella (2022).
<https://emerj.com/ai-sector-overviews/manufacturing-ai-use-cases-and-trends-an-executive-brief/>
- [9]. [Jagreet Kaur](#) (03 July 2023) | Machine Vision Working and Its Applications | A Quick Guide.
<https://www.xenonstack.com/blog/machine-vision-application>
- [10]. Machine Vision Market Size Worth \$18.24 Billion by 2025 (May 14, 2019).
https://www.bloomberg.com/press-releases/2019-05-14/machine-vision-market-size-worth-18-24-billion-by-2025-cagr-7-7-grand-view-research-inc?utm_source=website&utm_medium=share&utm_campaign=copy
- [11]. Zhou, Longfei; Zhang, Lin; Konz, Nicholas (2021): Computer Vision Techniques in Manufacturing. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.17125652.v2>
- [12]. itransition (23 May, 2023) Aleksandr Ahramovich | Computer vision in

manufacturing: 9 use cases, examples, and best practices.

<https://www.itransition.com/computer-vision/manufacturing>

[13]. pcspl (Feb 20, 2023) [precision](#) | Importance Of Vision Inspection In The Pharmaceutical Industry.

<https://www.pcspl.com/importance-of-vision-systems-for-inspection-pharmaceutical-companies/>

[14]. SENSUM (December 17, 2020) Visual inspection: The Reality Vs Ideal.

<https://www.sensum.eu/posts/visual-inspection-the-reality-vs-ideal/>

[15]. Canny, J., *A Computational Approach To Edge Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.

<http://dx.doi.org/10.1109/TPAMI.1986.4767851>

[16]. Ajitesh Kumar (August 2, 2022) | Perceptron Explained using Python Example.

<https://vitalflux.com/perceptron-explained-using-python-example/>

[17]. Bharath Ramsudar & Reza Bosagh Zadeh | TensorFlow for Deep Learning Chapter 4. Fully Connected Deep Networks.

<https://www.oreilly.com/library/view/tensorflow-for-deep/9781491980446/ch04.html>

[18]. Lecun, Y., Jackel, L. D., Bottou, L., Cortes, C., Denker, J. S., Drucker, H., Guyon, I., Muller, U. A., Sackinger, E., Simard, P., & Vapnik, V. (1995). Learning algorithms for classification: A comparison on handwritten digit recognition. In J. H. Oh, C. Kwon, & S. Cho (Eds.), *Neural networks: The statistical mechanics perspective* (pp. 261-276). World Scientific.

[19]. IndoML.com | Student Notes: Convolutional Neural Networks (CNN) Introduction.

<https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/>

[20]. Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba | Learning Deep Features for Discriminative Localization (2015).

<https://arxiv.org/abs/1512.04150>

[21]. P. Bergmann, M. Fauser, D. Sattlegger and C. Steger, "MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 9584-9592, doi: 10.1109/CVPR.2019.00982.

[22]. Karen Simonyan & Andrew Zisserman (2015) | Very Deep Convolutional Networks for Large-Scale Image Recognition.

<https://arxiv.org/abs/1409.1556>

[23]. Deng, Jia; Dong, Wei; Socher, Richard; Li, Li-Jia; Li, Kai; Fei-Fei, Li (2009),

["ImageNet: A Large-Scale Hierarchical Image Database"](#) (PDF), *2009 conference on*

Computer Vision and Pattern.

[25]. Srikanth Tammina (2019); Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images; International Journal of Scientific and Research Publications (IJSRP) 9(10) (ISSN: 2250-3153), DOI: <http://dx.doi.org/10.29322/IJSRP.9.10.2019.p9420>

[26]. Diederik P. Kingma & Jimmy Lei Ba (2015) | Adam: A Method for Stochastic Optimization. <https://arxiv.org/abs/1412.6980>

[27]. Min Lin, Qiang Chen, Shuicheng Yan (2013) | Network In Network. <https://arxiv.org/abs/1312.4400>

[28]. Jason Brownlee (May 6, 2019) | How to Visualize Filters and Feature Maps in Convolutional Neural Networks. <https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/>

[29]. SENSUM (May 21, 2020) | Reasons for most common tablet defects. <https://www.sensum.eu/posts/reasons-for-most-common-tablet-defects/>

[30]. Rheonics | Pharmaceutical printing process control. <https://rheonics.com/solutions-item/pharmaceutical-printing-process-control/>

[31]. Biogrund (2019) | Troubleshooting Guide. <https://www.biogrund.com/wp-content/uploads/2020/01/biogrund-troubleguide-a1poster-quer-11-2019.pdf>

[32]. Ashland (2019) | Film Coating Troubleshooting Guide. <http://filmcoating-troubleshooting.com/>

[33]. *John Norman* (24-Oct-2019) | Causes and solutions for dark spots in tablet manufacturing. https://www.manufacturingchemist.com/news/article_page/Causes_and_solutions_for_dark_spots_in_tablet_manufacturing/159326

[34]. G. Cole, J. Hogan, M. Aulton: Pharmaceutical Coating Technology, Taylor & Francis, 2002 <https://doi.org/10.3109/9780203014356>

[35]. [Bhowmik, Debjit. (2014). Tablet manufacturing processs and defects of tablets. 70. 24368-24374.

[36]. Daniel Godoy | Understanding binary cross-entropy / log loss: a visual explanation. <https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a>

[50] MobileNetV2: Inverted Residuals and Linear Bottlenecks 2018 | ([Mark Sandler](#), [Andrew Howard](#), [Menglong Zhu](#), [Andrey Zhmoginov](#), [Liang-Chieh Chen](#))
<https://doi.org/10.48550/arXiv.1801.04381>

[37] Keras Applications <https://keras.io/api/applications/#usage-examples-for-image-classification-models>

[38] Deep Residual Learning for Image Recognition 2015 | ([Kaiming He](#), [Xiangyu Zhang](#), [Shaoqing Ren](#), [Jian Sun](#))
<https://doi.org/10.48550/arXiv.1512.03385>

[39] Rethinking the Inception Architecture for Computer Vision 2015 | ([Christian Szegedy](#), [Vincent Vanhoucke](#), [Sergey Ioffe](#), [Jonathon Shlens](#), [Zbigniew Wojna](#))
<https://doi.org/10.48550/arXiv.1512.00567>

[40] EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks 2019 | ([Mingxing Tan](#), [Quoc V. Le](#))
<https://doi.org/10.48550/arXiv.1905.11946>

[41] Tensorflow Transfer Learning tutorial.
https://www.tensorflow.org/tutorials/images/transfer_learning#train_the_model