II

Πειραματικό Μέρος

Κεφάλαιο 3ο

Διεξαγωγή Πειραμάτων

3.1 Μεθοδολογία Πειραμάτων

Σε αυτή την ενότητα θα αναλυθεί η μεθοδολογία που ακολουθήθηκε κατά τη διάρκεια των πειραμάτων και θα παρουσιαστεί η λογική πίσω από τις συγκρίσεις που πραγματοποιήθηκαν. Η διεξαγωγή των πειραμάτων είχε ως στόχο να αξιολογηθεί η απόδοση των τριών διαφορετικών proxy tasks (Image Rotation Prediction, Image Colorization, Image Inpainting) μέσω της μεταφοράς μάθησης στο downstream task της ταξινόμησης εικόνων. Επιπλέον, θα εξηγήσουμε πώς οργανώθηκαν οι δοκιμές και η διαδικασία fine-tuning σε δύο διαφορετικά σύνολα δεδομένων (CIFAR-10 και CIFAR-100), και θα συζητηθεί η λογική των συγκρίσεων που πραγματοποιήθηκαν μεταξύ των διαφορετικών μεθόδων και παραμέτρων εκπαίδευσης, προκειμένου να διασφαλιστεί η δίκαιη και ακριβής αξιολόγηση των αποτελεσμάτων

3.1.1 Μεθοδολογία και Λογική Συγκρίσεων

Για να διασφαλίσουμε τη δίκαιη σύγκριση μεταξύ των τριών proxy tasks, επιλέξαμε το *Image Rotation Prediction*, το *Image Colorization* και το *Image Inpainting*. Αυτά τα τρία tasks είναι διαφορετικής φύσης: το πρώτο έχει να κάνει με τον προσανατολισμό των αντικειμένων μέσα στις εικόνες (rotation prediction), το δεύτερο αφορά τα χρώματα των εικόνων (colorization), ενώ το τρίτο εστιάζει στο περιεχόμενο, το πλαίσιο και τη συνοχή των εικόνων (inpainting). Χρησιμοποιήσαμε ως βάση το ResNet-50 για όλα τα tasks. Η ίδια αρχιτεκτονική εφαρμόστηκε σε κάθε task, προσαρμόζοντάς τη κατάλληλα ανάλογα με τις απαιτήσεις του κάθε proxy task. Σε επόμενη ενότητα θα αναλύσουμε ακριβώς πώς υλοποιήθηκε η προσαρμογή του ResNet-50 για κάθε ένα από αυτά τα tasks. Αυτή η ομοιομορφία στην αρχιτεκτονική επιτρέπει την ακριβή σύγκριση των επιδόσεων, χωρίς να επηρεάζονται τα αποτελέσματα από διαφοροποιήσεις στην αρχιτεκτονική του δικτύου.

Η λογική των συγκρίσεων εστιάζει σε δύο βασικές στρατηγικές fine-tuning:

1. Fine-tuning μόνο του τελευταίου επιπέδου: Σε αυτή την περίπτωση, επανεκπαιδεύσαμε μόνο το τελευταίο πλήρως συνδεδεμένο επίπεδο του ResNet-50 στο downstream task της ταξινόμησης εικόνων. Ο στόχος ήταν να δούμε πώς συγκρίνονται τα βάρη που αποκτήθηκαν από τα proxy tasks σε σχέση με τυχαία βάρη (*random weights*), επιτρέποντας μας να αξιολογήσουμε τι πληροφορία έχουν μάθει τα proxy tasks και αν αυτή η πληροφορία είναι χρήσιμη στο downstream task. Η σύγκριση με τα τυχαία βάρη μας δίνει τη δυνατότητα να αξιολογήσουμε κατά πόσο τα βάρη που έμαθε το δίκτυο μέσω του self-supervised learning είναι συμβατά και ωφέλιμα για την ταξινόμηση εικόνων.
2. Fine-tuning ολόκληρου του μοντέλου: Σε αυτή την περίπτωση, επανεκπαιδεύσαμε όλα τα επίπεδα του μοντέλου στο downstream task. Ο στόχος εδώ είναι να δούμε αν η εκπαίδευση των proxy tasks πραγματικά αξίζει να γίνει, βοηθώντας το μοντέλο να βελτιώσει την απόδοσή του σε σχέση με την αρχικοποίηση με τυχαία βάρη. Σημειώνουμε ότι χρησιμοποιήθηκαν οι ίδιες υπερπαράμετροι σε όλες τις περιπτώσεις για να διασφαλιστεί ότι η μόνη διαφορά ανάμεσα στα πειράματα ήταν η αρχική κατάσταση των βαρών.

Δεν χρησιμοποιήσαμε καθόλου *data augmentation* στα πειράματά μας, καθώς αυτό θα έθετε σε κίνδυνο τη δίκαιη σύγκριση. Κάθε proxy task έχει διαφορετικές απαιτήσεις σχετικά με τους μετασχηματισμούς (*transforms*), γεγονός που θα μπορούσε να προκαλέσει προβλήματα κατά την εκπαίδευση. Για παράδειγμα, στο *image rotation prediction task*, εάν εφαρμοστεί ένα *random rotation* 180° σε μια εικόνα κατά την

προεπεξεργασία δεδομένων και της αποδοθεί η κλάση 2 (180° περιστροφή), το ίδιο *random rotation* απο τους μετασχηματισμούς μπορεί να την επαναφέρει στην αρχική της θέση. Αυτό θα την έκανε να ανήκει ξανά στην κλάση 0, όμως εμείς θα της είχαμε δώσει λανθασμένα την κλάση 2. Ακόμη και αν το μοντέλο κάνει τη σωστή πρόβλεψη για την κλάση 0, το λάθος στις ετικέτες θα το καταγράψει ως λάθος πρόβλεψη, γεγονός που θα καταστρέψει την εκπαίδευση. Αντίστοιχα προβλήματα προκύπτουν και σε tasks όπως το *colorization*, όπου ο μετασχηματισμός *color jitter* θα μπορούσαν να αλλοιώσουν τα χρώματα, καθιστώντας δύσκολη την ακριβή εκπαίδευση. Σε κανονικές συνθήκες, μπορούμε να χρησιμοποίησαμε κανονικά τους μετασχηματισμούς (εφόσον δεν καταστρέφουν την εκπαίδευση της έμμεσης διεργασίας) για να επιτύχουμε υψηλότερη ακρίβεια ωστόσο όπως αναφέραμε σκοπός ήταν η δίκαιη σύγκριση και όχι η υψηλότερη δυνατή επίδοση. Για αυτούς τους λόγους, περιοριστήκαμε σε βασικά *transforms*, όπως *resize*, *normalize* και *totensor*. Σημαντικό είναι να τονίσουμε ότι επειδή ο στόχος μας δεν ήταν να πετύχουμε την υψηλότερη δυνατή επίδοση, αλλά να εξασφαλίσουμε δίκαιη σύγκριση των μεθόδων, οι επιδόσεις που θα καταγραφούν θα είναι πιθανώς ελάχιστα χαμηλότερες από τα αναμενόμενα standards.

Τα proxy tasks εκπαιδεύτηκαν στο CIFAR-100 χωρίς τη χρήση των labels του dataset, καθώς το χρησιμοποιήσαμε ως *unlabeled* σύνολο δεδομένων. Στη συνέχεια, τα αποθηκευμένα βάρη από τα proxy tasks μεταφέρθηκαν και τεσταρίστηκαν τόσο στο CIFAR-10 όσο και στο CIFAR-100 για το downstream task του *image classification*. Η χρήση και των δύο datasets ήταν σκόπιμη, καθώς θέλαμε να αξιολογήσουμε την απόδοση τόσο στο ίδιο dataset που χρησιμοποιήθηκε για την εκπαίδευση των proxy tasks (CIFAR-100) όσο και σε ένα διαφορετικό αλλά παρόμοιο dataset (CIFAR-10). Τα δύο σύνολα δεδομένων επιλέχθηκαν λόγω της ομοιότητάς τους, καθώς έχουν το ίδιο μέγεθος και περιέχουν εικόνες παρόμοιας φύσης, γεγονός που επιτρέπει την πιο ομαλή σύγκριση των αποτελεσμάτων.

Εκτός από τη σύγκριση των τελικών επιδόσεων των proxy tasks, εξετάσαμε επίσης το πόσο γρήγορα κάθε proxy task επιτυγχάνει υψηλή απόδοση. Δηλαδή, μας ενδιαφέρει να δούμε όχι μόνο ποιο task αποδίδει καλύτερα, αλλά και ποιο μπορεί να φτάσει σε ένα ικανοποιητικό επίπεδο ακρίβειας σε λιγότερες εποχές εκπαίδευσης. Για παράδειγμα, εάν ένα μοντέλο που έχει εκπαιδευτεί με *SSL* μπορεί να φτάσει σε ικανοποιητικό επίπεδο ακρίβειας με λιγότερες εποχές σε σχέση με τυχαία βάρη, τότε αυτό θα ήταν οικονομικά και χρονικά πιο αποδοτικό. Τα διαγράμματα που απεικονίζουν τη διαφορά στην απόδοση μεταξύ των proxy task weights και των random weights μας βοηθούν να αξιολογήσουμε αυτήν τη διάσταση.

Τέλος, πρέπει να σημειωθεί ότι δεν πραγματοποιήθηκε εκτενές *hyperparameter tuning* στο downstream task ανάλογα με την χρήση των διαφορετικών βαρών από τα διαφορετικά tasks. Η λογική μας ήταν να διατηρήσουμε τις υπερπαραμέτρους σταθερές σε όλα τα πειράματα για να διασφαλίσουμε τη δίκαιη σύγκριση. Αυτό σημαίνει ότι δεν επιδιώξαμε να πιάσουμε τις καλύτερες δυνατές επιδόσεις, αλλά να δημιουργήσουμε ένα ισοδύναμο πειραματικό περιβάλλον για τη σύγκριση των μεθόδων. Επομένως, οι επιδόσεις των μοντέλων δεν ανταγωνίζονται τα βέλτιστα πρότυπα απόδοσης, αλλά αποσκοπούν στην απόδειξη της αποτελεσματικότητας των proxy tasks σε σχέση με τα random weights και στην ανάδειξη της πιο αποδοτικής διεργασίας μεταξύ των proxy tasks ως προ-εκπαιδευτική μέθοδος των βαρών για το downstream task της ταξινόμησης εικόνων.

3.1.2 Ερωτήματα και Προσδοκίες

Με βάση τη μεθοδολογία που αναλύθηκε στην προηγούμενη ενότητα, τα κύρια ερευνητικά ερωτήματα και προσδοκίες των πειραμάτων μας συνοψίζονται ως εξής:

Επιδόσεις με τη χρήση Αυτο-Εποπτευόμενης Μάθησης σε σχέση με τα τυχαία βάρη

Αρχικό και βασικό μας ερώτημα είναι εάν η χρήση των proxy tasks μέσω της αυτο-εποπτευόμενης μάθησης (*self-supervised learning*) μπορεί να προσφέρει καλύτερα αποτελέσματα από τα τυχαία βάρη. Αναμένουμε ότι σε σχεδόν όλες τις μετρήσεις, οι επιδόσεις θα είναι καλύτερες με τη χρήση των προεκπαιδευμένων βαρών, καθώς αυτά θα έχουν μάθει κάποια βασικά χαρακτηριστικά από τα proxy tasks, σε αντίθεση με τα

τυχαία βάρη που δεν φέρουν καμία πληροφορία.

Επιδόσεις στα διαφορετικά datasets (CIFAR-10 vs CIFAR-100)

Αναμένουμε ότι οι επιδόσεις στο CIFAR-100 θα είναι χαμηλότερες σε σύγκριση με το CIFAR-10, καθώς το CIFAR-100 έχει 100 κλάσεις, ενώ το CIFAR-10 έχει μόνο 10. Αυτό σημαίνει ότι το task ταξινόμησης είναι πιο πολύπλοκο στο CIFAR-100. Ωστόσο, περιμένουμε ότι τα pretrained weights που εκπαιδεύτηκαν στο CIFAR-100 θα λειτουργήσουν εξίσου καλά και στο CIFAR-10, καθώς τα δύο datasets περιέχουν παρόμοιες εικόνες και χαρακτηριστικά. Επομένως, αναμένουμε ότι τα proxy tasks θα είναι ικανά να μεταφέρουν τις γνώσεις τους στο νέο dataset και να διατηρήσουν καλές αποδόσεις.

Fine-tuning μόνο στο τελευταίο επίπεδο

Σε αυτή την περίπτωση, το ερώτημα είναι κατά πόσο το fine-tuning μόνο του τελευταίου επιπέδου του δικτύου θα αποφέρει καλύτερα αποτελέσματα με τα proxy tasks σε σχέση με τα τυχαία βάρη. Αναμένουμε ότι οι επιδόσεις των proxy tasks θα είναι σημαντικά καλύτερες, καθώς τα βάρη του δικτύου έχουν ήδη εκπαιδευτεί σε ένα συναφές task. Εδώ, τα proxy tasks παρέχουν προεκπαιδευμένα βάρη που περιέχουν χρήσιμη πληροφορία, ενώ τα τυχαία βάρη απαιτούν να μάθουν τα πάντα από το μηδέν. Επομένως, αφού εκπαιδεύουμε μόνο το τελευταίο επίπεδο, και δεδομένου ότι τα τυχαία βάρη δεν έχουν υποστεί εκπαίδευση εκτός από το τελευταίο επίπεδο, ενώ τα προεκπαιδευμένα βάρη περιέχουν ήδη κάποια σχετική πληροφορία στο μεγαλύτερο ποσοστό του μοντέλου, αναμένουμε σημαντικά καλύτερες επιδόσεις από τα proxy tasks.

Fine-tuning σε ολόκληρο το μοντέλο

Το ερώτημα εδώ είναι εάν το fine-tuning ολόκληρου του μοντέλου με τα προεκπαιδευμένα βάρη θα είναι καλύτερο από το fine-tuning ολόκληρου του μοντέλου με τυχαία βάρη. Προφανώς, περιμένουμε να πετύχουμε πολύ καλύτερες επιδόσεις σε σύγκριση με το fine-tuning μόνο του τελευταίου επιπέδου, καθώς σε αυτή την περίπτωση επανεκπαιδεύονται όλα τα επίπεδα του δικτύου και όχι μόνο το τελευταίο. Όσον αφορά το fine-tuning ολόκληρου του μοντέλου, αναμένουμε ότι οι τελικές επιδόσεις με τα pre-trained weights θα είναι ελάχιστα καλύτερες από αυτές με τα random weights, καθώς τα βάρη είναι μόνο τα αρχικά - είτε προεκπαιδευμένα είτε τυχαία - και στη συνέχεια όλα τροποποιούνται κατά τη διάρκεια της εκπαίδευσης. Επομένως, περιμένουμε μία ελαφρώς καλύτερη σύγκλιση (convergence) στο τέλος με τα pre-trained weights, αλλά η διαφορά δεν θα είναι δραματική.

Ταχύτητα επίτευξης ικανοποιητικής επίδοσης με fine-tuning όλων των επιπέδων του μοντέλου

Εκτός από τις τελικές επιδόσεις, μας ενδιαφέρει και η ταχύτητα με την οποία τα proxy tasks θα βοηθήσουν το μοντέλο να φτάσει σε ένα ικανοποιητικό επίπεδο απόδοσης. Αναμένουμε ότι τα proxy tasks θα επιτρέψουν στο μοντέλο να πετύχει γρήγορα υψηλά επίπεδα ακρίβειας. Πιστεύουμε ότι σε πολύ λιγότερες εποχές από ό,τι με τα τυχαία βάρη, το μοντέλο θα φτάσει σε υψηλή ακρίβεια.

Έμμεση διεργασία με τις καλύτερες επιδόσεις

Ένα βασικό ερώτημα που θέλουμε να απαντήσουμε είναι ποιο από τα τρία proxy tasks (Image Rotation Prediction, Image Colorization, Image Inpainting) θα αποδειχθεί πιο αποτελεσματικό στην ενίσχυση της απόδοσης του downstream task. Περιμένουμε ότι το *colorization* ή το *inpainting* θα αποδειχθούν τα πιο αποδοτικά, καθώς είναι πιο σύνθετες διεργασίες σε σχέση με την πρόβλεψη περιστροφής, και επομένως προσφέρουν περισσότερη πληροφορία στο μοντέλο.

3.2 Υλοποίηση Έμμεσων Διεργασιών

Στην ενότητα αυτή, θα παρουσιαστεί η υλοποίηση τριών έμμεσων διεργασιών που χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου μέσω αυτο-εποπτευόμενης μάθησης: η πρόβλεψη περιστροφής εικόνας, η χρωματοποίηση εικόνας, και η επιδιόρθωση/συμπλήρωση εικόνας (image inpainting). Θα αναλύσουμε τον τρόπο με τον οποίο υλοποιήθηκαν αυτά τα proxy/pretext tasks, τα εργαλεία που χρησιμοποιήθηκαν για την κατασκευή τους, καθώς και τις υπερπαραμέτρους που επιλέχθηκαν για την εκπαίδευση των μοντέλων σε αυτά τα tasks. Επιπλέον θα παρουσιαστούν εικόνες που δείχνουν την επεξεργασία των αρχικών εικόνων και τις προβλέψεις του εκάστοτε μοντέλου.
Οι φωτογραφίες που χρησιμοποιήθηκαν για την απεικόνιση της απόδοσης των μοντέλων δεν προέρχονται από το σύνολο δεδομένων που χρησιμοποιήθηκε στα πειράματα, το οποίο θα περιγραφεί σε επόμενη ενότητα. Αυτό συμβαίνει επειδή οι εικόνες του συγκεκριμένου συνόλου δεδομένων είναι πολύ μικρού μεγέθους, γεγονός που δυσκολεύει την οπτική αντίληψη των διαφορών από το ανθρώπινο μάτι. Αντ' αυτού, χρησιμοποιήθηκαν εικόνες από το σύνολο δεδομένων Caltech-256 [52], το οποίο περιλαμβάνει εικόνες με επαρκείς διαστάσεις, επιτρέποντας έτσι την καλύτερη παρατήρηση και κατανόηση των διαφορών μεταξύ των εικόνων.

3.2.1 Υλοποίηση Διεργασίας Πρόβλεψης Περιστροφής Εικόνας

Προεπεξεργασία Δεδομένων: Η διαδικασία προεπεξεργασίας δεδομένων στην διεργασία πρόβλεψης περιστροφής εικόνας περιλαμβάνει αρχικά την αναπροσαρμογή των εικόνων σε σταθερό μέγεθος 224x224 pixels (μετασχηματισμός "*Resize*()"), ώστε να διασφαλιστεί η ομοιομορφία των εισόδων στο μοντέλο. Επίσης οι εικόνες υφίστανται μετασχηματισμούς, όπως η μετατροπή τους σε tensors (μετασχηματισμός "*ToTensor*" - οι τιμές των pixels μετατρέπονται από το εύρος [0, 255] σε [0,1], μας επιστρέφεται ο τένσορας με διαστασεις [Κανάλια, Ύψος, Πλάτος]), καθώς και κανονικοποίηση (μετασχηματισμός "*Normalize(mean, std)*" - κανονικοποιεί τα δεδομένα εισόδου έτσι ώστε η εικόνα να έχει συγκεκριμένη μέση τιμή(mean) και τυπική απόκλιση (std)) κάτι που επιτρέπει την αποτελεσματική επεξεργασία τους από τα επίπεδα του νευρωνικού δικτύου.

Αυτή η διαδικασία προετοιμάζει τα δεδομένα ώστε να είναι έτοιμα για εκπαίδευση, ενώ η τυχαία περιστροφή σε κάθε εποχή διασφαλίζει ότι το μοντέλο εκτίθεται σε διαφορετικές γωνίες της ίδιας εικόνας, ενισχύοντας έτσι την ικανότητά του να μάθει τις χωρικές δομές των εικόνων.Στις εικόνες που ανακτώνται από τον φάκελο, εφαρμόζεται τυχαία περιστροφή σε γωνίες 0°, 90°, 180° ή 270°.

Ο κύριος στόχος είναι η ταξινόμηση των περιστροφών, όπου κάθε εικόνα ανήκει σε μία από τις τέσσερις κατηγορίες, ανάλογα με τη γωνία περιστροφής της. Η τιμή της ετικέτας καθορίζεται από τον τύπο: *ετικέτα = γωνία περιστροφής / 90*, δίνοντας τιμές ετικέτας 0, 1, 2 ή 3 για τις αντίστοιχες περιστροφές 0°, 90°, 180° και 270°. Έτσι, οι 4 κατηγορίες προκύπτουν από τις διαφορετικές γωνίες, και το μοντέλο πρέπει να μάθει να αναγνωρίζει σωστά την περιστροφή κάθε εικόνας. Θα μπορούσαμε να πούμε ότι αναγάγουμε τη συγκεκριμένη διεργασία σε μια διεργασία ταξινόμησης εικόνας, αλλά με ετικέτες που υποδηλώνουν τη γωνία περιστροφής αντί για το περιεχόμενο της εικόνας όπως στην κλασική διεργασία ταξινόμησης εικόνων.

Αρχιτεκτονική: Η αρχιτεκτονική που χρησιμοποιήθηκε είναι το μοντέλο *ResNet50* που αποτελείται από 50 στρώσεις (layers), οι οποίες περιλαμβάνουν συνελικτικά στρώματα (convolutional layers), στρώματα pooling, και πλήρως συνδεδεμένα στρώματα. Οι τελευταίες στρώσεις είναι πλήρως συνδεδεμένες, με την τελική να έχει 1000 εξόδους για προβλήματα ταξινόμησης στο *ImageNet*.

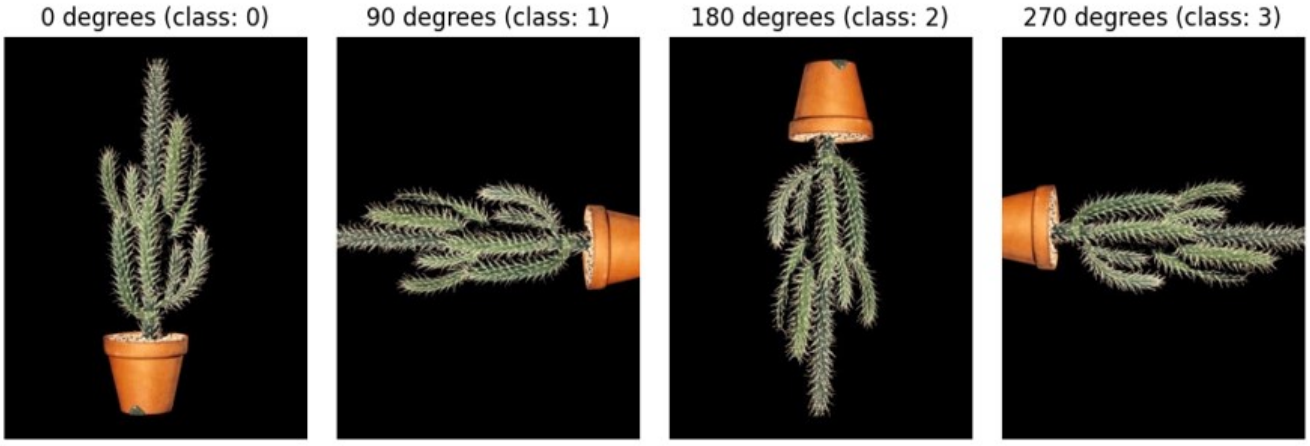
Στο συγκεκριμένο task της πρόβλεψης περιστροφής εικόνας, αυτή η τελική στρώση τροποποιήθηκε ώστε να έχει 4 εξόδους, μία για κάθε πιθανή γωνία περιστροφής (0°, 90°, 180°, 270°). Το υπόλοιπο του δικτύου

παραμένει το ίδιο και προσαρμόζεται ώστε να εξάγει χαρακτηριστικά που βοηθούν στην πρόβλεψη της σωστής γωνίας.

Εκπαίδευση: Η εκπαίδευση του μοντέλου για το task της πρόβλεψης περιστροφής εικόνας υλοποιήθηκε με τη χρήση του *CrossEntropyLoss*, το οποίο είναι κατάλληλο για προβλήματα ταξινόμησης όπως αυτό. Το *CrossEntropyLoss* μετρά τη διαφορά μεταξύ της προβλεπόμενης κατανομής πιθανοτήτων του μοντέλου και των πραγματικών ετικετών. Ειδικά για το συγκεκριμένο task, οι τέσσερις γωνίες περιστροφής (0°, 90°, 180°, 270°) αντιστοιχούν σε τέσσερις κατηγορίες, και το *CrossEntropyLoss* είναι ιδανικό για ταξινομήσεις πολλαπλών κατηγοριών.

Η επιλογή του Adam optimizer έγινε γιατί παρέχει γρήγορη σύγκλιση και δυναμική προσαρμογή του learning rate, βελτιστοποιώντας το μοντέλο με τη χρήση των παραγώγων της συνάρτησης απώλειας για την ενημέρωση των βαρών. Το learning rate έχει οριστεί στο 0.001 ώστε να διατηρεί τη σταθερότητα του μοντέλου, ενώ το weight decay στο 0.0001 βοηθάει στην αποτροπή της υπερπροσαρμογής μειώνοντας την πολυπλοκότητα του μοντέλου.

Το μοντέλο εκπαιδεύεται για 50 εποχές με 64 παρτίδες (batches), λαμβάνοντας δεδομένα που περιστρέφονται σε τυχαίες γωνίες και προσπαθώντας να προβλέψει σωστά την γωνία κάθε φορά.



Εικόνα 3.2.1 - 1: Παραδείγματα περιστροφής εικόνας για το task του Image Rotation Prediction (Caltech-256 Dataset): Η αρχική εικόνα περιστρέφεται κατά 0°, 90°, 180°, και 270°, με κάθε γωνία να αντιστοιχεί σε μια κλάση (0, 1, 2, 3).

3.2.2 Υλοποίηση Διεργασίας Χρωματοποίησης Εικόνας

Προεπεξεργασία Δεδομένων: Αρχικά χρησιμοποιούνται τα ίδια transforms όπως και στα υπόλοιπα tasks. Αυτά περιλαμβάνουν τη μετατροπή του μεγέθους των εικόνων σε σταθερές διαστάσεις με το *Resize*, την κανονικοποίηση των τιμών των pixel με το *Normalize*, και τη μετατροπή της εικόνας σε μορφή tensor μέσω του *ToTensor*, ώστε να είναι κατάλληλη για εισαγωγή στο νευρωνικό δίκτυο.

Στη συνέχεια, για να δημιουργηθούν τα κατάλληλα δεδομένα εισόδου για το μοντέλο, οι εικόνες μετατρέπονται από έγχρωμες (RGB) σε ασπρόμαυρες (*grayscale*). Η μετατροπή αυτή είναι απαραίτητη καθώς το task του *image colorization* στοχεύει στο να εκπαιδεύσει το μοντέλο να "χρωματίσει" μια ασπρόμαυρη εικόνα. Για να μπορέσει όμως αυτή η ασπρόμαυρη εικόνα να εισαχθεί στο δίκτυο, χρειάζεται να ταιριάξει τη μορφή της με αυτή των έγχρωμων εικόνων, δηλαδή να έχει τρία κανάλια. Γι’ αυτό επαναλαμβάνουμε την *grayscale* εικόνα τρεις φορές, δημιουργώντας ένα *fake RGB* format. Κάθε ένα από τα τρία κανάλια της νέας εικόνας έχει την ίδια πληροφορία με την *grayscale* εικόνα, έτσι ώστε η είσοδος να έχει τη σωστή διαστασιολόγηση για το νευρωνικό δίκτυο, που είναι σχεδιασμένο να δουλεύει με εικόνες

τριών καναλιών.

Τέλος, κάθε δείγμα δεδομένων που επιστρέφεται περιέχει δύο στοιχεία: την ασπρόμαυρη εικόνα που χρησιμοποιείται ως είσοδος στο μοντέλο, και την έγχρωμη εικόνα ως στόχο, την οποία το μοντέλο καλείται να ανακατασκευάσει με την κατάλληλη χρωματική πληροφορία.

Αρχιτεκτονική: Η αρχιτεκτονική του δικτύου που χρησιμοποιείται για το task του image colorization, είναι μια μορφή autoencoder με βάση το ResNet50. Αρχικά, αφαιρούνται τα τελικά πλήρως συνδεδεμένα επίπεδα (fully connected layers) του ResNet, καθώς αυτά είναι σχεδιασμένα για ταξινόμηση και δεν είναι κατάλληλα για το task της χρωματοποίησης εικόνας. Αντικαθίστανται με διαδοχικά επίπεδα αποκωδικοποίησης (decoder), που βασίζονται σε transposed convolutions.

Το δίκτυο λειτουργεί με τον εξής τρόπο: Υπάρχουν 5 επίπεδα του encoder και 5 επίπεδα του decoder. Κατά την κωδικοποίηση (encoder), τα χαρακτηριστικά της εικόνας περνούν μέσα από τα συνελικτικά επίπεδα του ResNet, που εξάγουν πληροφορίες υψηλού επιπέδου από την εικόνα. Στη συνέχεια, κατά την αποκωδικοποίηση (decoder), το μοντέλο χρησιμοποιεί transposed convolutions για να επαναφέρει το μέγεθος της εικόνας σε κανονικό επίπεδο. Πιο συγκεκριμένα, τα transposed convolutions, γνωστά και ως deconvolutions ή upsampling convolutions, είναι μια διαδικασία που αντιστρέφει τη λειτουργία των κανονικών συνελικτικών επιπέδων. Αντί να μειώνουν το μέγεθος των χαρακτηριστικών της εικόνας, όπως κάνουν τα κανονικά convolutions, τα transposed convolutions αυξάνουν το μέγεθος, επαναφέροντας τη διαστασιολογία της εικόνας στα αρχικά της επίπεδα. Αυτό τα καθιστά ιδιαίτερα χρήσιμα στα decoders, όπως στα autoencoders, όπου το μοντέλο χρειάζεται να ανακατασκευάσει εικόνες από μειωμένες αναπαραστάσεις. Σε κάθε επίπεδο του decoder εφαρμόζονται λειτουργίες BatchNorm2D και ReLU για κανονικοποίηση και εισαγωγή μη-γραμμικότητας, που συμβάλλουν στη σταθεροποίηση της εκπαίδευσης και τη βελτίωση των προβλέψεων.

Σημαντικό ρόλο παίζουν οι συνδέσεις παράκαμψης (skip connections) μεταξύ των επιπέδων του encoder και του decoder. Αυτές οι συνδέσεις διασφαλίζουν ότι οι πληροφορίες από τα αρχικά στάδια της επεξεργασίας της εικόνας δεν χάνονται κατά την αποκωδικοποίηση. Έτσι, το μοντέλο καταφέρνει να αναπαραγάγει λεπτομέρειες και υφές με μεγαλύτερη ακρίβεια.

Το τελικό επίπεδο του μοντέλου είναι υπεύθυνο για την παραγωγή της ανακατασκευασμένης εικόνας, διασφαλίζοντας ότι το αποτέλεσμα έχει το ίδιο μέγεθος και τα ίδια χαρακτηριστικά με την αρχική εικόνα.

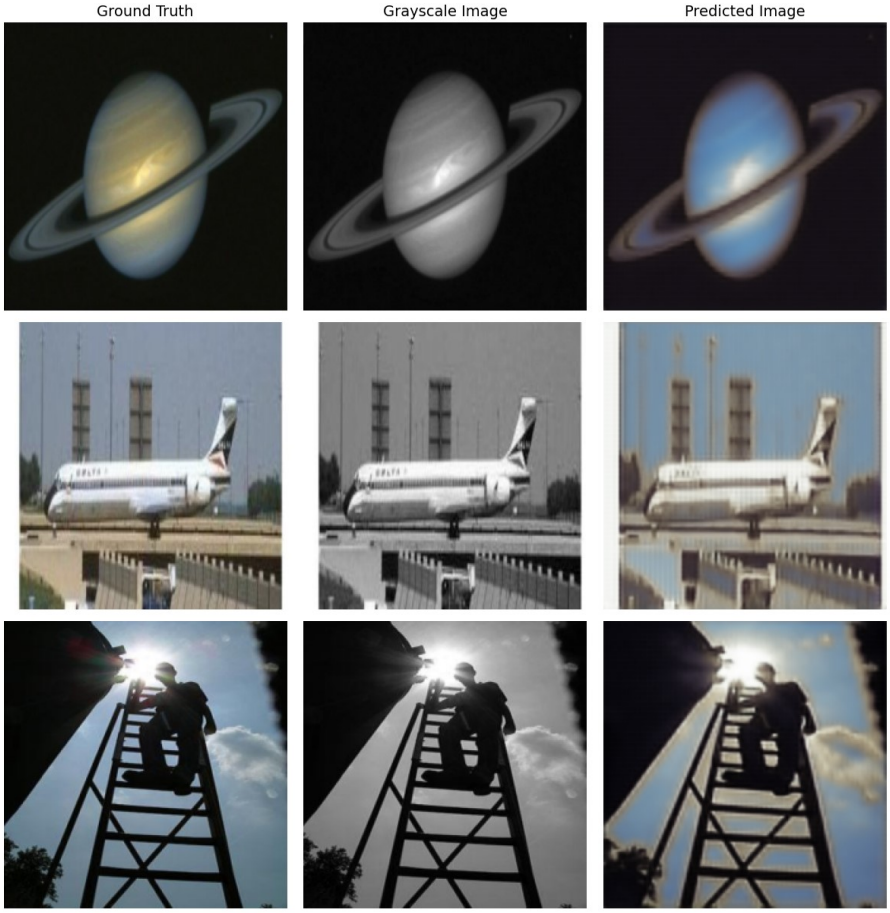
Εκπαίδευση: Για το task του *image colorization*, η διαδικασία εκπαίδευσης επικεντρώνεται στην εκμάθηση του μοντέλου να ανακατασκευάζει ολόκληρη την εικόνα, προσθέτοντας τη σωστή χρωματική πληροφορία στα ασπρόμαυρα δεδομένα που του παρέχονται ως είσοδος. Κατά τη διάρκεια της εκπαίδευσης, γίνεται χρήση της συνάρτησης απώλειας MSE Loss (Mean Squared Error), η οποία μετράει το τετραγωνικό σφάλμα μεταξύ των προβλεπόμενων χρωμάτων και των πραγματικών τιμών των pixels σε όλη την εικόνα.

Σε κάθε βήμα της εκπαίδευσης, το μοντέλο λαμβάνει ως είσοδο μια ασπρόμαυρη εικόνα και προσπαθεί να παράγει την αντίστοιχη έγχρωμη εκδοχή της. Η συνάρτηση MSE Loss υπολογίζει τη διαφορά ανάμεσα στις προβλεπόμενες τιμές των pixels και στις πραγματικές τιμές των χρωμάτων (RGB) σε ολόκληρη την εικόνα. Σε αντίθεση με το *image inpainting* που θα δούμε παρακάτω, όπου η απώλεια (loss) υπολογίζεται μόνο για τα καλυμμένα τμήματα, εδώ η MSE Loss εφαρμόζεται σε κάθε pixel της εικόνας, καθώς το μοντέλο πρέπει να χρωματίσει πλήρως όλες τις περιοχές της εικόνας.

Η επιλογή της MSE Loss είναι ιδανική για το *image colorization* επειδή τα δεδομένα της εικόνας είναι συνεχείς τιμές (τιμές pixels), και η MSE είναι κατάλληλη για την ελαχιστοποίηση των διαφορών μεταξύ της πρόβλεψης και της πραγματικής εικόνας. Στοχεύει στο να προσαρμόζει το χρώμα σε κάθε pixel της εικόνας με τέτοιο τρόπο ώστε οι προβλεπόμενες τιμές να πλησιάζουν όσο το δυνατόν περισσότερο τις πραγματικές τιμές χρώματος.

Κατά την εκπαίδευση, το μοντέλο περνά από πολλαπλά βήματα, γνωστά ως εποχές, στα οποία επεξεργάζεται εικόνες από το *training set*. Το σύνολο δεδομένων χωρίζεται σε *batches* με μέγεθος 64, και το μοντέλο εκτελεί επαναλαμβανόμενες ενημερώσεις των παραμέτρων του χρησιμοποιώντας τον βελτιστοποιητή Adam, με ρυθμό μάθησης 0.001. Κατά τη διάρκεια κάθε epoch (50 συνολικά), το μοντέλο εκπαιδεύεται και η απώλεια (loss) υπολογίζεται και συσσωρεύεται για να παρακολουθείται η συνολική απόδοση.

Παρακάτω παρατίθενται ορισμένες προβλέψεις του μοντέλου σε εικόνες του Caltech-256 Dataset (για την καλύτερη κατανόηση των διαφορών των εικόνων), οι οποίες απεικονίζουν τη διαδικασία χρωματοποίησης των ασπρόμαυρων εικόνων (ορισμένες από τις προβλέψεις του μοντέλου είναι επιτυχημένες, ενώ άλλες δεν αποδίδουν το χρωματικό αποτέλεσμα με την ίδια ακρίβεια):



Εικόνα 3.2.2 - 1: Παραδείγματα από το task της χρωματοποίησης εικόνων (image colorization). Στην πρώτη στήλη εμφανίζονται οι αρχικές έγχρωμες εικόνες (Ground Truth), στη δεύτερη στήλη οι ασπρόμαυρες εκδοχές των εικόνων (Grayscale Image) και στην τρίτη στήλη οι προβλέψεις του μοντέλου για την επαναφορά των χρωμάτων (Predicted Image).

3.2.3 Υλοποίηση Διεργασίας Επιδιόρθωσης/Συμπλήρωσης Εικόνας

Προεπεξεργασία Δεδομένων: Στην υλοποίηση της διαδικασίας για το image inpainting task, αρχικά κάθε εικόνα προετοιμάζεται με τη μετατροπή της σε ομοιόμορφο μέγεθος και μορφή που μπορεί να επεξεργαστεί το μοντέλο μετασχηματίζοντας τις διαστάσεις της εικόνας σε 224x224 (Resize), κανονικοποιώντας την (Normalize) και μετατρέποντας την σε τένσορα (ToTensor), ακριβώς όπως και στα

προηγούμενα tasks. Η επεξεργασία περιλαμβάνει την εφαρμογή μιας δυαδικής μάσκας στην εικόνα, η οποία αποτελείται από τιμές 0 και 1, όπου τα 0 αναπαριστούν τις περιοχές που θέλουμε να καλύψουμε και να συμπληρώσει το μοντέλο, και τα 1 αναπαριστούν τις περιοχές που παραμένουν ακέραιες.

Η μάσκα δημιουργείται τυχαία με τη χρήση γεωμετρικών σχημάτων, όπως γραμμές και κύκλοι, που τοποθετούνται σε τυχαία σημεία στην εικόνα. Αυτό έχει ως αποτέλεσμα την κάλυψη ορισμένων περιοχών της εικόνας, τις οποίες το μοντέλο καλείται να ανακατασκευάσει. Το επόμενο βήμα είναι ο πολλαπλασιασμός της εικόνας με τη μάσκα, δημιουργώντας την "masked" εικόνα.

Σε κάθε εποχή εκπαίδευσης, η μάσκα εφαρμόζεται τυχαία, δηλαδή κάθε εικόνα μπορεί να λάβει διαφορετική μάσκα σε κάθε επανάληψη. Αυτό βοηθά το μοντέλο να εκπαιδευτεί πάνω σε πολλές παραλλαγές της ίδιας εικόνας, βελτιώνοντας την ικανότητά του να γενικεύει και να συμπληρώνει σωστά τα κενά.

Αρχιτεκτονική: Για το task του image inpainting, χρησιμοποιείται η ίδια αρχιτεκτονική με αυτή που περιγράφηκε για το image colorization, δηλαδή ένα autoencoder βασισμένο στο ResNet50. Η κύρια διαφορά είναι ότι το μοντέλο εστιάζει στην επιδιόρθωση των κενών περιοχών της εικόνας όπως θα περιγράψουμε και παρακάτω στην εκπαίδευση, ενώ για την ανακατασκευή χρησιμοποιούνται τα ίδια επίπεδα αποκωδικοποίησης με transposed convolutions και οι συνδέσεις παράκαμψης (skip connections) για διατήρηση των λεπτομερειών.

Εκπαίδευση: Κατά την εκπαίδευση του image inpainting, έγινε χρήση της συνάρτησης MSE Loss (Mean Squared Error), η οποία είναι ιδανική για τη συγκεκριμένη περίπτωση, καθώς στοχεύει στην ελαχιστοποίηση της διαφοράς ανάμεσα στις προβλεπόμενες και πραγματικές τιμές των pixels μόνο στα καλυμμένα μέρη της εικόνας (masked regions). Η MSE Loss υπολογίζει το τετραγωνικό σφάλμα μεταξύ των προβλεπόμενων και πραγματικών τιμών των pixels. Σε αυτό το task, όμως, η απώλεια (loss) υπολογίζεται μόνο για τα καλυμμένα (masked) μέρη της εικόνας, δηλαδή τα τμήματα που έχουν αλλοιωθεί σκόπιμα.

Η υπόλοιπη εικόνα, που παραμένει άθικτη, δεν επηρεάζεται από το μοντέλο. Ο σκοπός είναι το μοντέλο να μάθει να ανακατασκευάζει μόνο τα χαμένα τμήματα, διατηρώντας τις δομές και τις λεπτομέρειες της αρχικής εικόνας. Έτσι, με τη χρήση της MSE Loss για τα masked κομμάτια, το μοντέλο μαθαίνει να παράγει αποτελέσματα που προσεγγίζουν όσο το δυνατόν περισσότερο τα πραγματικά δεδομένα στα καλυμμένα σημεία, χωρίς να επηρεάζει τις υπόλοιπες περιοχές.

Η επιλογή της MSE είναι κατάλληλη, καθώς τα δεδομένα της εικόνας είναι συνεχής πληροφορία, και η MSE εξασφαλίζει την ελαχιστοποίηση της διαφοράς μεταξύ της πρόβλεψης και της πραγματικής εικόνας με τρόπο που λαμβάνει υπόψη τις τιμές των pixels και τις μεταβολές τους.

Συνοψίζοντας, η διαδικασία εκπαίδευσης για το image inpainting ακολουθεί τα εξής βήματα: Κάθε εικόνα αφού πολλαπλασιαστεί με μια τυχαία παραγόμενη μάσκα, η οποία καλύπτει ορισμένα τμήματα της εικόνας έχει μία masked μορφή. Η masked εικόνα χρησιμοποιείται ως είσοδος στο μοντέλο, το οποίο προσπαθεί να προβλέψει τα καλυμμένα σημεία. Η πρόβλεψη συγκρίνεται μόνο με τα masked μέρη της αρχικής εικόνας, ενώ η υπόλοιπη εικόνα παραμένει αμετάβλητη. Η διαδικασία αυτή εκτελείται για 50 epochs με batch size 64, εξασφαλίζοντας σταδιακή βελτίωση της ικανότητας του μοντέλου να ανακατασκευάζει τα χαμένα τμήματα. Για την βελτιστοποίηση χρησιμοποιήθηκε ο Adam Optimizer με learning rate ίσο με 0.0001. Σε κάθε epoch, μια διαφορετική τυχαία μάσκα εφαρμόζεται σε κάθε εικόνα, γεγονός που επιτρέπει στο μοντέλο να βλέπει την ίδια εικόνα με πολλές διαφορετικές απώλειες δεδομένων (μάσκες). Αυτό ενισχύει την ικανότητά του να γενικεύει και να προσαρμόζεται σε διαφορετικά σενάρια όπου τμήματα της εικόνας λείπουν.

Στη συνέχεια, παρουσιάζονται κάποιες ενδεικτικές προβλέψεις του μοντέλου σε εικόνες του Caltech-256 Dataset (για την καλύτερη αντίληψη των διαφορών των εικόνων) όπου φαίνεται η επιδιόρθωση/ συμπλήρωση των τυχαίων masked περιοχών των εικόνων (μερικές από τις προβλέψεις του μοντέλου είναι

πετυχημένες, ενώ σε άλλες περιπτώσεις η αποκατάσταση των κενών είναι λιγότερο ακριβής):

Εικόνα 3.2.3 - 1: Παραδείγματα εικόνων από το task της επιδιόρθωσης/συμπλήρωσης εικόνας (image inpainting). Στην πρώτη στήλη απεικονίζεται η αρχική εικόνα (Ground Truth), στη δεύτερη στήλη παρουσιάζεται η εικόνα με τα καλυμμένα τμήματα (Masked Image), και στην τρίτη στήλη η εικόνα που προβλέφθηκε από το μοντέλο (Predicted Image).

3.3 Downstream Tasks και Σύνολα Δεδομένων

Τα downstream tasks αποτελούν το επόμενο στάδιο όπου αξιολογείται η απόδοση των προεκπαιδευμένων μοντέλων σε πραγματικές εφαρμογές, όπως η ταξινόμηση εικόνων. Στην περίπτωση αυτή, το μοντέλο που εκπαιδεύτηκε σε proxy tasks, όπως η πρόβλεψη περιστροφής, η χρωματοποίηση ή η συμπλήρωση εικόνας, δοκιμάζεται στο downstream task της ταξινόμησης εικόνων. Αυτό το στάδιο είναι κρίσιμο, καθώς αναδεικνύει κατά πόσο τα χαρακτηριστικά που έμαθε το μοντέλο από τα proxy tasks μπορούν να γενικευτούν σε πιο πρακτικές και απαιτητικές εφαρμογές. Επιπλέον, η επιλογή των κατάλληλων συνόλων δεδομένων είναι εξίσου σημαντική, καθώς αυτά επηρεάζουν άμεσα την απόδοση και τη δυνατότητα γενίκευσης του μοντέλου. Στην ενότητα αυτή θα εξηγηθούν τόσο το task της ταξινόμησης εικόνων όσο και τα σύνολα δεδομένων που χρησιμοποιήθηκαν.

3.3.1 Ταξινόμηση Εικόνων ως Downstream Task

Η Ταξινόμηση Εικόνων (*Image Classification*) είναι ένα κλασικό πρόβλημα στην υπολογιστική όραση, το οποίο στοχεύει στην ανάθεση μιας εικόνας σε μία από πολλές προκαθορισμένες κατηγορίες. Το task αυτό απαιτεί την ικανότητα του μοντέλου να εξάγει αποδοτικά χαρακτηριστικά από την εικόνα και να τα συσχετίζει με τις αντίστοιχες κλάσεις. Η ταξινόμηση εικόνων λειτουργεί ως downstream task σε πολλά ερευνητικά πειράματα, επιτρέποντας την αξιολόγηση της γενίκευσης των χαρακτηριστικών που έχουν μάθει τα μοντέλα από προγενέστερα proxy tasks.

Στο πλαίσιο αυτό, το μοντέλο εκπαιδεύεται με στόχο να αναγνωρίσει υψηλού επιπέδου χαρακτηριστικά, όπως σχήματα, υφές και χρώματα, τα οποία είναι κρίσιμα για την ορθή κατηγοριοποίηση των εικόνων. Η προσέγγιση αυτή στηρίζεται στις αρχές της πολυεπίπεδης εκμάθησης (*hierarchical learning*), όπου τα πρώτα επίπεδα του νευρωνικού δικτύου εξάγουν απλά χαρακτηριστικά, όπως ακμές, ενώ τα επόμενα επίπεδα συνδυάζουν αυτά τα χαρακτηριστικά για να σχηματίσουν πιο σύνθετες αναπαραστάσεις.

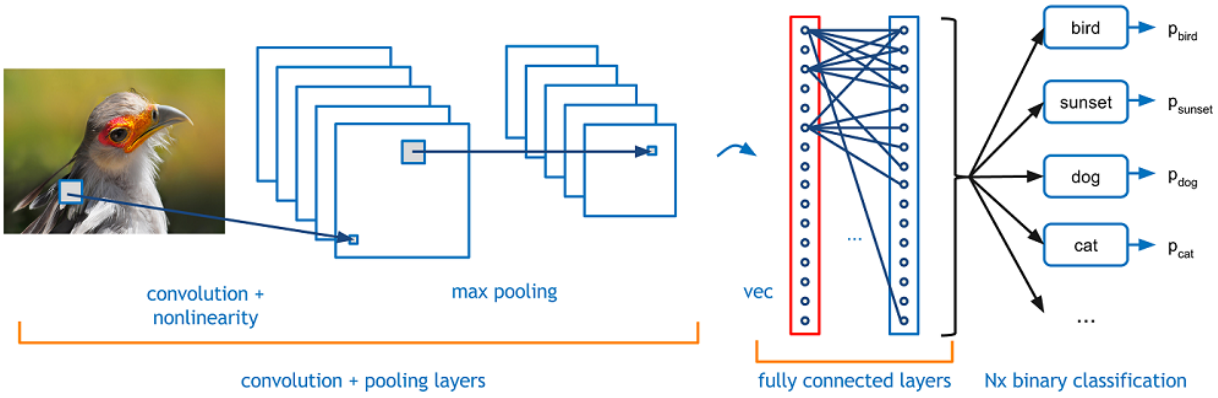
Η διαδικασία της εκπαίδευσης περιλαμβάνει 50 εποχές και το μέγεθος παρτίδας είναι 64. Επιπλέον, ως συνάρτηση απώλειας χρησιμοποιήθηκε η Cross-Entropy Loss και για τη βελτιστοποίηση χρησιμοποιήσαμε τον Adam Optimizer με learning rate ίσο με 0.001 και με weight decay ίσο με 0.0001 ως μία τεχνική κανονικοποίησης. Τέλος, χρησιμοποιήθηκε η τεχνική Dropout όπου κρίθηκε απαραίτητο για την αποφυγή της υπερπροσαρμογής.

Η απόδοση του μοντέλου στο task της ταξινόμησης εικόνων αξιολογείται μέσω μετρικών όπως η ακρίβεια (*accuracy*), η οποία αντικατοπτρίζει το ποσοστό των σωστών προβλέψεων σε σχέση με το σύνολο των εικόνων. Μια υψηλή ακρίβεια υποδηλώνει ότι το μοντέλο έχει μάθει να αναγνωρίζει σωστά τις κλάσεις, ακόμα και όταν οι εικόνες είναι οπτικά περίπλοκες ή περιέχουν θόρυβο. Στο συγκεκριμένο downstream task, η δυνατότητα του μοντέλου να εκμεταλλευτεί τα χαρακτηριστικά που έμαθε κατά την προεκπαίδευση μέσω άλλων έμμεσων διεργασιών είναι κρίσιμη για την τελική του απόδοση.

Τα νευρωνικά δίκτυα που χρησιμοποιούνται για ταξινόμηση εικόνων, όπως το ResNet, εφαρμόζουν συνήθως διαδοχικές συνελικτικές (*convolutional*) και πλήρως συνδεδεμένες (*fully connected*) στρώσεις. Τα τελικά επίπεδα του δικτύου εξάγουν μια αναπαράσταση της εικόνας σε έναν χώρο χαρακτηριστικών υψηλής διάστασης και εφαρμόζεται μια συνάρτηση ενεργοποίησης (π.χ. *softmax*) για την εκτίμηση της πιθανότητας ότι η εικόνα ανήκει σε κάθε κατηγορία. Η κατηγορία με την υψηλότερη πιθανότητα επιλέγεται ως η τελική πρόβλεψη του μοντέλου.

Η ταξινόμηση εικόνων ως downstream task επιτρέπει την αξιολόγηση του πώς τα προεκπαιδευμένα βάρη και τα χαρακτηριστικά που μαθαίνει το μοντέλο από proxy tasks μπορούν να συμβάλλουν στη γενίκευση και εφαρμογή αυτών των χαρακτηριστικών σε πιο συγκεκριμένες και απαιτητικές εργασίες υπολογιστικής

όρασης.



Εικόνα 3.3.1 - 1: Διαδικασία του task της ταξινόμησης εικόνων, όπου ένα συνελικτικό νευρωνικό δίκτυο (CNN) επεξεργάζεται μια εικόνα μέσω συνελικτικών και max pooling επιπέδων, για να εξάγει χαρακτηριστικά και να πραγματοποιήσει ταξινόμηση σε προκαθορισμένες κατηγορίες σύμφωνα με τις παραγόμενες πιθανότητες P. (Πηγή: TowardsDataScience)

3.3.2 Σύνολα Δεδομένων που Χρησιμοποιήθηκαν

Στα πειράματα χρησιμοποιήθηκαν τα σύνολα δεδομένων CIFAR-10 και CIFAR-100, τα οποία είναι από τα πιο ευρέως χρησιμοποιούμενα datasets στον τομέα της υπολογιστικής όρασης. Αυτά τα σύνολα δεδομένων περιέχουν έγχρωμες εικόνες μικρών διαστάσεων (32x32 pixels), και έχουν ως στόχο την ταξινόμηση εικόνων σε πολλαπλές κατηγορίες.

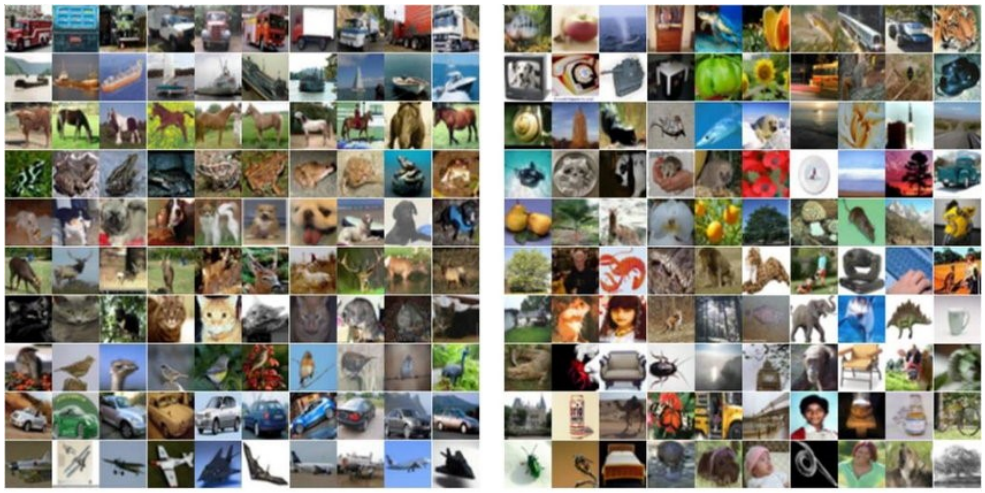
Το CIFAR-10 περιέχει συνολικά 60.000 εικόνες, κατανεμημένες σε 10 διαφορετικές κλάσεις. Κάθε κλάση περιέχει 6.000 εικόνες, εκ των οποίων 5.000 χρησιμοποιούνται για την εκπαίδευση και 1.000 για την αξιολόγηση του μοντέλου. Οι κλάσεις περιλαμβάνουν διάφορα αντικείμενα όπως αυτοκίνητα, γάτες, αεροπλάνα, σκύλους, και άλλα καθημερινά αντικείμενα και ζώα. Το CIFAR-10 χρησιμοποιείται συχνά σε tasks ταξινόμησης εικόνων, καθώς οι εικόνες του είναι απλές και ευρέως κατανοητές, ενώ η μικρή τους διάσταση επιτρέπει ταχύτερη εκπαίδευση.

Το CIFAR-100 είναι παρόμοιο με το CIFAR-10, αλλά περιέχει 100 κατηγορίες αντί για 10, καθιστώντας το task της ταξινόμησης πολύ πιο απαιτητικό. Κάθε κλάση περιλαμβάνει 600 εικόνες, με 500 για την εκπαίδευση και 100 για την αξιολόγηση. Οι κατηγορίες του CIFAR-100 είναι πιο εξειδικευμένες και περιλαμβάνουν υποκατηγορίες όπως έντομα, ψάρια, έπιπλα, και άλλα καθημερινά αντικείμενα. Παρά τη μεγαλύτερη ποικιλία κατηγοριών, οι εικόνες του CIFAR-100 έχουν τις ίδιες διαστάσεις (32x32 pixels) και παρόμοια πολυπλοκότητα με το CIFAR-10.

Για το training των proxy tasks, χρησιμοποιήθηκε το CIFAR-100. Τα χαρακτηριστικά που έμαθε το μοντέλο κατά τη διάρκεια της εκπαίδευσης στα proxy tasks με αυτό το dataset μεταφέρθηκαν στο downstream task της ταξινόμησης εικόνων. Για την ταξινόμηση, τα pre-trained βάρη από το CIFAR-100 δοκιμάστηκαν τόσο στο ίδιο dataset (CIFAR-100) όσο και σε ένα διαφορετικό dataset, το CIFAR-10, προκειμένου να αξιολογηθεί η ικανότητα του μοντέλου να γενικεύει σε διαφορετικά σύνολα δεδομένων.

Τα CIFAR-10 και CIFAR-100 είναι συμβατά μεταξύ τους, καθώς έχουν παρόμοια χαρακτηριστικά. Και τα δύο datasets περιέχουν εικόνες με την ίδια ανάλυση (32x32) και έχουν ένα ευρύ φάσμα κατηγοριών αντικειμένων. Αυτή η συμβατότητα μεταξύ των συνόλων δεδομένων επιτρέπει τη δοκιμή των προεκπαιδευμένων βαρών από το CIFAR-100 στο CIFAR-10, παρέχοντας χρήσιμες πληροφορίες σχετικά με

το κατά πόσο τα χαρακτηριστικά που έμαθε το μοντέλο μπορούν να γενικευτούν στο task ταξινόμησης.



Εικόνα 3.3.2 - 1: (a) Εικόνες από το CIFAR-10 και (b) εικόνες από το CIFAR-100, παρουσιάζοντας παραδείγματα κατηγοριών που περιέχονται στα datasets. (Πηγή: ResearchGate)

3.4 Μεταφορά Μάθησης και Fine-Tuning

Η Μεταφορά Μάθησης (*Transfer learning*) και το Fine-Tuning αποτελούν κρίσιμες τεχνικές στη σύγχρονη μηχανική μάθηση, ιδιαίτερα σε περιπτώσεις όπου το σύνολο δεδομένων για το downstream task είναι περιορισμένο. Η βασική ιδέα της μεταφοράς μάθησης είναι η χρήση ενός προεκπαιδευμένου μοντέλου, το οποίο έχει μάθει χρήσιμα χαρακτηριστικά από ένα διαφορετικό task, με στόχο να βελτιώσει την απόδοση σε ένα νέο, αλλά συναφές task. Το fine-tuning συνίσταται στην περαιτέρω προσαρμογή των προεκπαιδευμένων βαρών του μοντέλου στο νέο task, είτε προσαρμόζοντας μόνο τα τελευταία επίπεδα του δικτύου είτε επανεκπαιδεύοντας ολόκληρο το μοντέλο. Στην παρούσα ενότητα, θα εξεταστούν οι στρατηγικές μεταφοράς μάθησης από τα proxy tasks στο downstream task της ταξινόμησης εικόνων, καθώς και οι τεχνικές fine-tuning που εφαρμόστηκαν για τη βελτιστοποίηση της απόδοσης του μοντέλου

3.4.1 Μεταφορά Μάθησης από τα Proxy Tasks στο Downstream Task

Η διαδικασία της μεταφοράς μάθησης από τα proxy tasks στο downstream task βασίζεται στη χρήση των βέλτιστων pretrained weights που αποκτήθηκαν κατά την εκπαίδευση στις τρεις έμμεσες διεργασίες (proxy tasks): πρόβλεψη περιστροφής, χρωματοποίηση και επιδιόρθωση/συμπλήρωση εικόνας (inpainting), οι οποίες εκπαιδεύτηκαν στο σύνολο δεδομένων CIFAR-100. Είναι σημαντικό να σημειωθεί ότι το CIFAR-100 χρησιμοποιήθηκε ως ένα unlabeled dataset για την εκπαίδευση των proxy tasks, χωρίς να χρησιμοποιηθούν τα labels των εικόνων που υποδηλώνουν τις κατηγορίες στις οποίες ανήκουν. Κατά τη διάρκεια της εκπαίδευσης στα proxy tasks, παρακολουθήσαμε τη βελτίωση του validation accuracy και, κάθε φορά που το μοντέλο σημείωνε την υψηλότερη απόδοση στο validation set, αποθηκεύαμε το state του μοντέλου. Αυτό εξασφάλισε ότι διατηρήσαμε τα πιο αποδοτικά και γενικεύσιμα βάρη, απαραίτητα για τη μεταφορά στο downstream task.

Αφού ολοκληρώθηκε η εκπαίδευση στα proxy tasks, κατασκευάσαμε ένα πανομοιότυπο μοντέλο για το downstream task της ταξινόμησης εικόνων. Για να μπορέσουμε να χρησιμοποιήσουμε τα αποθηκευμένα βάρη, φορτώσαμε τα pretrained weights στο νέο μοντέλο και στη συνέχεια τροποποιήσαμε το τελευταίο πλήρως συνδεδεμένο επίπεδο (fully connected layer), έτσι ώστε να προσαρμοστεί στον αριθμό των κλάσεων που απαιτούσε το εκάστοτε dataset. Συγκεκριμένα, για το σύνολο δεδομένων CIFAR-10, το τελικό επίπεδο αναδιαμορφώθηκε ώστε να παράγει 10 εξόδους, όσες είναι οι κατηγορίες στο CIFAR-10, ενώ για το CIFAR-100 το τελικό επίπεδο τροποποιήθηκε ώστε να παράγει 100 εξόδους.

Η προσαρμογή αυτή του τελευταίου επιπέδου είναι σημαντική, καθώς επιτρέπει τη σωστή αντιστοίχιση του χώρου χαρακτηριστικών που έχει μάθει το μοντέλο με τον αριθμό των κλάσεων που απαιτούνται από το εκάστοτε dataset. Με τη διαδικασία αυτή, μπορέσαμε να αξιολογήσουμε την αποτελεσματικότητα των προεκπαιδευμένων βαρών που αποκτήθηκαν από το CIFAR-100 όχι μόνο στο ίδιο dataset αλλά και σε ένα διαφορετικό dataset, το CIFAR-10. Αυτό μας επιτρέπει να εξετάσουμε κατά πόσο τα χαρακτηριστικά που έμαθε το μοντέλο σε ένα σύνολο δεδομένων μπορούν να γενικευτούν και να βελτιώσουν την απόδοση σε ένα διαφορετικό σύνολο δεδομένων.

3.4.2 Fine-Tuning στο Downstream Task

Στη διαδικασία του fine-tuning στο downstream task της ταξινόμησης εικόνων, πραγματοποιήσαμε δύο διαφορετικές προσεγγίσεις για την εκπαίδευση του μοντέλου. Συγκεκριμένα, εκπαιδεύσαμε το μοντέλο σε κάθε σύνολο δεδομένων (CIFAR-10 και CIFAR-100) με δύο τρόπους:

1. Εκπαίδευση ολόκληρου του μοντέλου, προσαρμόζοντας τα βάρη όλων των επιπέδων (fine-tuning all layers).
2. Εκπαίδευση μόνο του τελευταίου επιπέδου του δικτύου (fine-tuning only the last layer), ενώ διατηρήσαμε τα υπόλοιπα επίπεδα "παγωμένα" (frozen).

Στην πρώτη περίπτωση, πραγματοποιήθηκε πλήρης εκπαίδευση του δικτύου σε κάθε task, επιτρέποντας στα βάρη όλων των επιπέδων να ενημερώνονται κατά τη διαδικασία της εκπαίδευσης. Στη δεύτερη περίπτωση, για να επανεκπαιδευτεί μόνο το τελευταίο επίπεδο, παγώσαμε (freeze) όλα τα επίπεδα του δικτύου και “ξεπαγώσαμε” (unfreeze) το τελευταίο επίπεδο, αφήνοντας ελεύθερο προς εκπαίδευση μόνο το τελευταίο πλήρως συνδεδεμένο επίπεδο. Με αυτόν τον τρόπο, μπορέσαμε να αξιολογήσουμε πώς τα προεκπαιδευμένα βάρη επηρεάζουν την απόδοση του μοντέλου, όταν γίνεται fine-tuning μόνο στο τελευταίο επίπεδο.

Αυτή η διαδικασία εφαρμόστηκε σε όλα τα proxy tasks (πρόβλεψη περιστροφής, χρωματοποίηση και επιδιόρθωση/συμπλήρωση εικόνας). Ο στόχος ήταν να αξιολογήσουμε την απόδοση του μοντέλου σε διαφορετικά σύνολα δεδομένων και με διαφορετικές στρατηγικές fine-tuning.

Για την προεπεξεργασία των δεδομένων, χρησιμοποιήσαμε τους εξής μετασχηματισμούς: αλλαγή μεγέθους των εικόνων σε 224x224 pixels, μετατροπή τους σε tensors και κανονικοποίηση (normalize) των pixel values με τις τιμές μέσου όρου και τυπικής απόκλισης που αντιστοιχούν στο CIFAR-10 και CIFAR-100. Η χρήση της κανονικοποίησης εξασφαλίζει ότι τα δεδομένα εισόδου έχουν σταθεροποιημένες τιμές, κάτι που βελτιώνει την εκπαίδευση του μοντέλου.

Οι υπερπαράμετροι της εκπαίδευσης ήταν κοινές για όλα τα tasks, τα μοντέλα των οποίων εκπαιδεύτηκαν για 50 εποχές με το batch size να ορίζεται σε 64. Επίσης, ο αλγόριθμος βελτιστοποίησης ήταν ο Adam, με learning rate 0.001 και weight decay 1e-4. Η εκπαίδευση πραγματοποιήθηκε με χρήση της συνάρτησης απώλειας CrossEntropyLoss, καθώς το downstream task αφορά ταξινόμηση πολλών κατηγοριών.

Με τη στρατηγική αυτή, μπορέσαμε να αξιολογήσουμε την απόδοση των προεκπαιδευμένων βαρών σε διαφορετικά σύνολα δεδομένων, τόσο με πλήρες fine-tuning όσο και με fine-tuning μόνο του τελευταίου επιπέδου. Η διαδικασία που περιγράφηκε, εφαρμόστηκε ακριβώς με τον ίδιο τρόπο για τα βάρη όλων των proxy tasks ώστε να διασφαλιστεί η δίκαιη σύγκριση μεταξύ των έμμεσων διεργασιών.

3.5 Αποτελέσματα

Σε αυτή την ενότητα θα παρουσιαστούν τα αποτελέσματα των πειραμάτων που πραγματοποιήθηκαν με σκοπό την αξιολόγηση των αποδόσεων των προεκπαιδευμένων βαρών στο task της ταξινόμησης εικόνων.

3.5.1 Αποτελέσματα 1ου πειράματος

Task: Image Classification

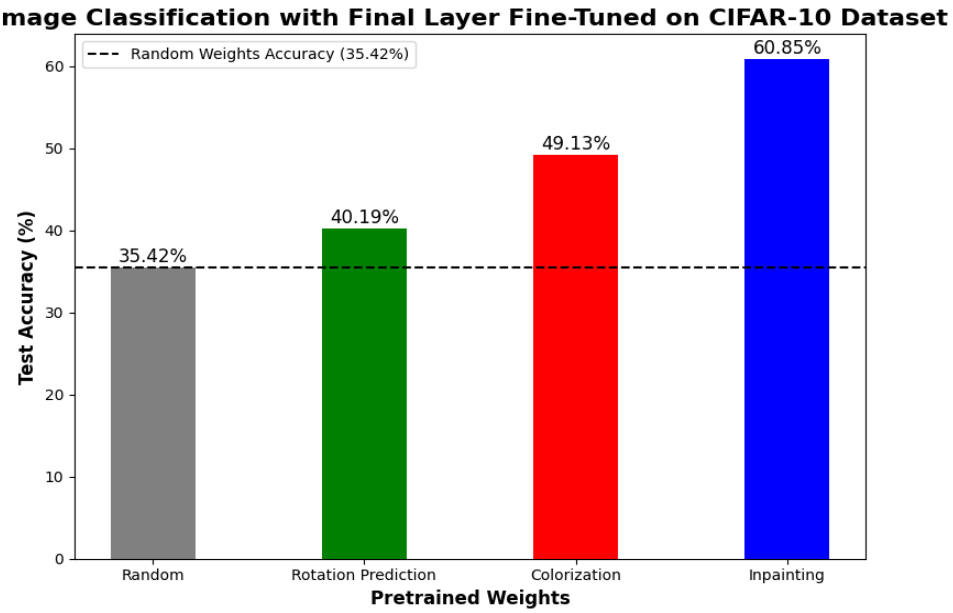
Fine-Tuning: Final Layer

Weights: Randomly Initialized, Image Rotation Prediction Pretraining, Image Colorization Pretraining, Image Inpainting Pretraining

Dataset: CIFAR-10

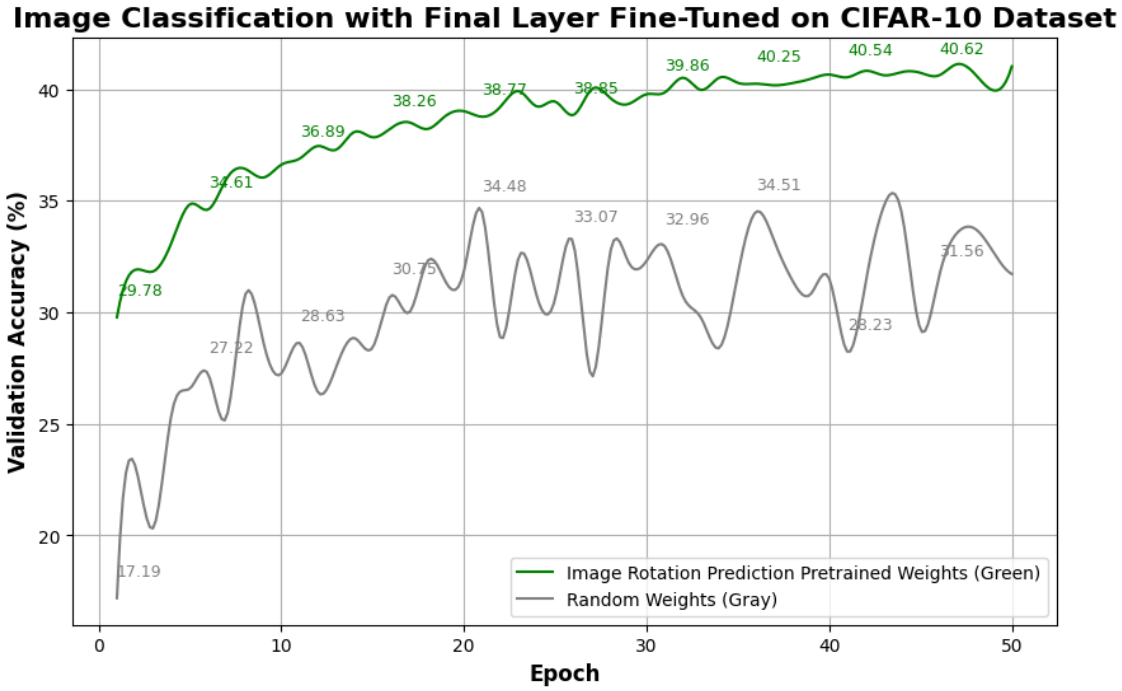
Accuracy Type: Test Accuracy

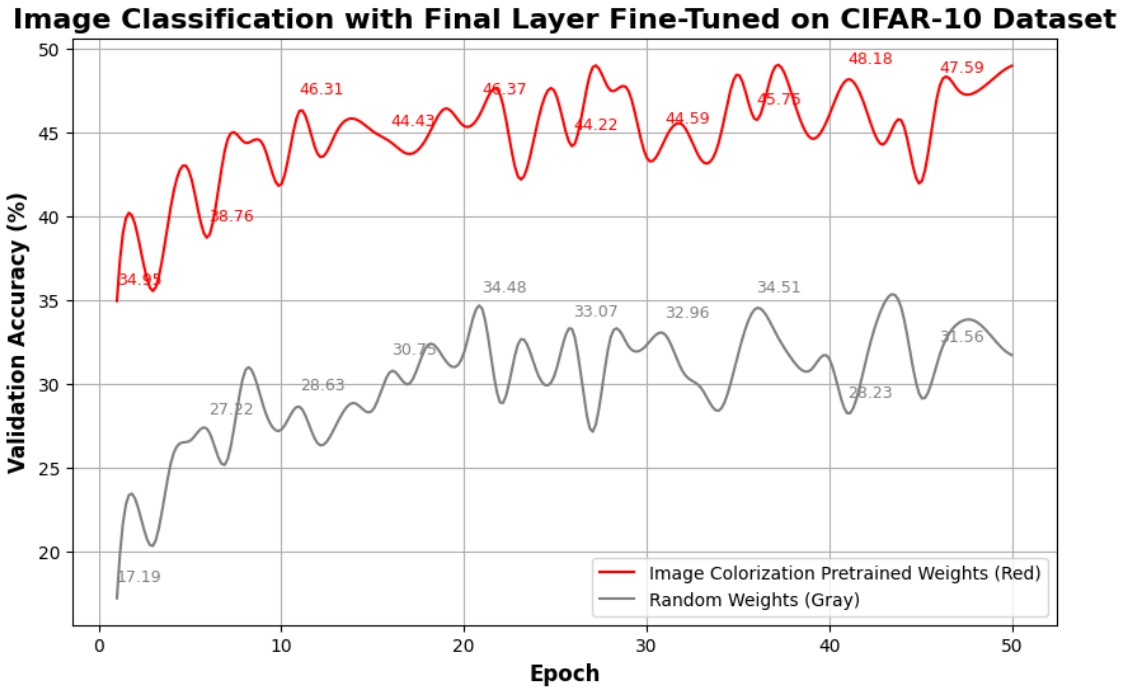
Weights	Test Accuracy
Randomly Initialized	35.42%
Image Rotation Prediction Pretraining	40.19%
Image Colorization Pretraining	49.13%
Image Inpainting Pretraining	60.85%

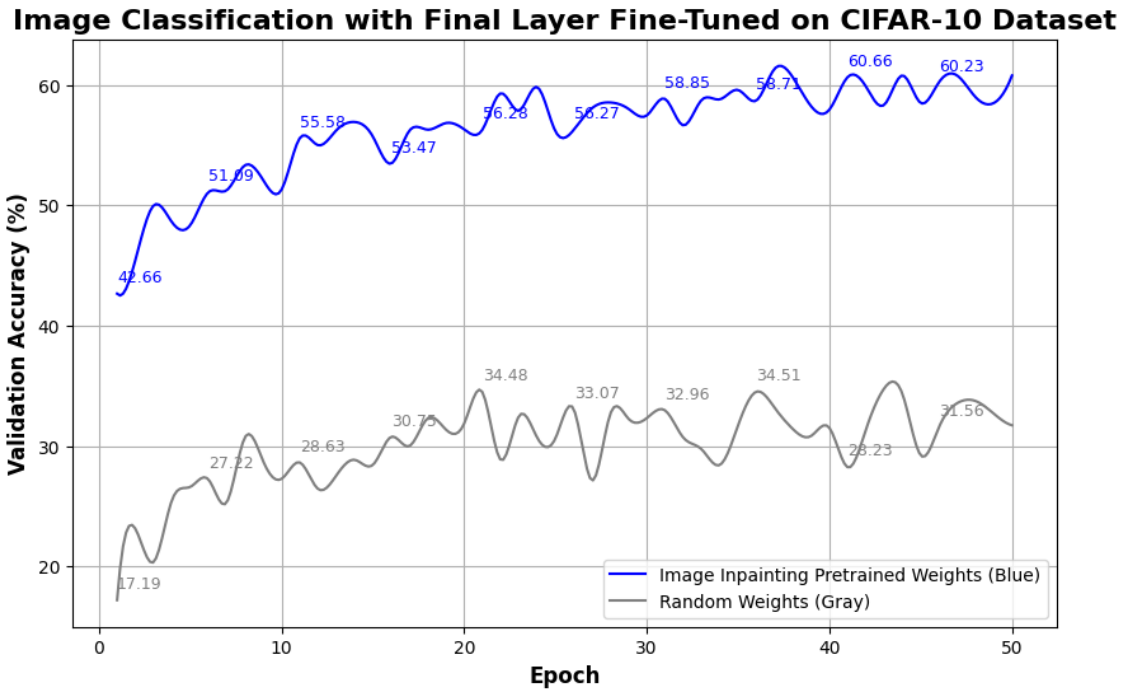


Πίνακας-Διάγραμμα 3.5.1 - 1: Αποτελέσματα ακρίβειας (Test Accuracy) στην ταξινόμηση εικόνων, επανεκπαιδεύοντας μόνο το τελευταίο στρώμα του μοντέλου στη συγκεκριμένη διεργασία, χρησιμοποιώντας τυχαία και προεκπαιδευμένα βάρη από τις τρεις έμμεσες διεργασίες (Image Rotation Prediction, Image Colorization, Image Inpainting) στο σύνολο δεδομένων CIFAR-10.

Αφού παρατέθηκαν οι τελικές τιμές Test Accuracy για το πρώτο πείραμα, στη συνέχεια θα παρουσιαστούν πιο συγκεκριμένες μετρήσεις για το ίδιο πείραμα, αναλύοντας τις τιμές του Validation Accuracy ανά εποχή. Αυτές οι μετρήσεις θα μας δώσουν μια πιο λεπτομερή εικόνα για την απόδοση του μοντέλου κατά τη διάρκεια της εκπαίδευσης:







Διαγράμματα 3.5.1 - 2,3,4: Αποτελέσματα ακρίβειας (στο validation set) ανά εποχή στην ταξινόμηση εικόνων, επανεκπαιδεύοντας μόνο το τελευταίο στρώμα του μοντέλου στη συγκεκριμένη διεργασία, χρησιμοποιώντας τυχαία βάρη ως σημείο αναφοράς και τα προεκπαιδευμένα βάρη από τις τρεις έμμεσες διεργασίες: Image Rotation Prediction, Image Colorization, Image Inpainting αντίστοιχα, στο σύνολο δεδομένων CIFAR-10.

3.5.2 Αποτελέσματα 2ου πειράματος

Task: Image Classification

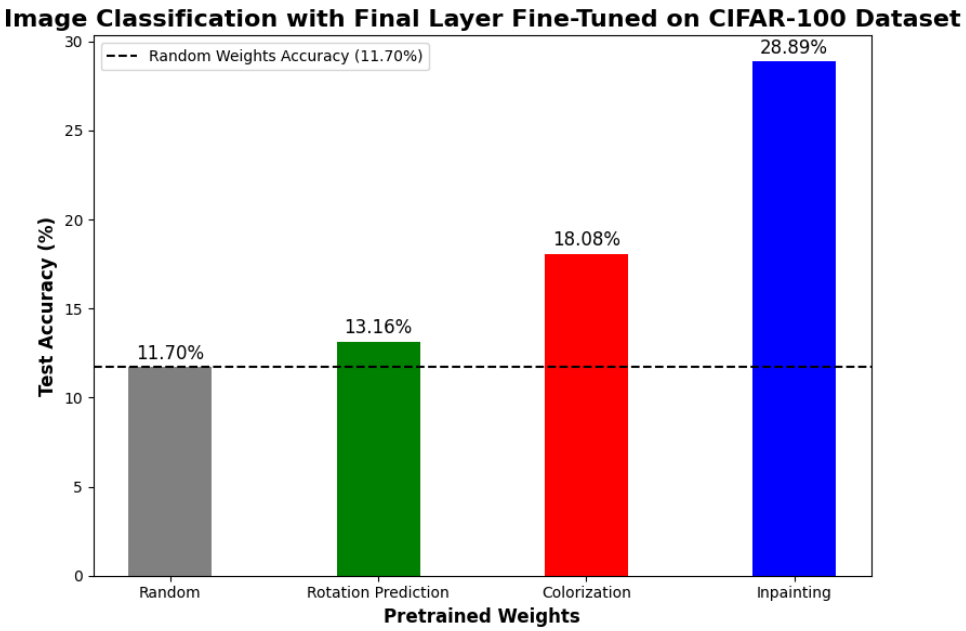
Fine-Tuning: Final Layer

Weights: Randomly Initialized, Image Rotation Prediction Pretraining, Image Colorization Pretraining, Image Inpainting Pretraining

Dataset: CIFAR-100

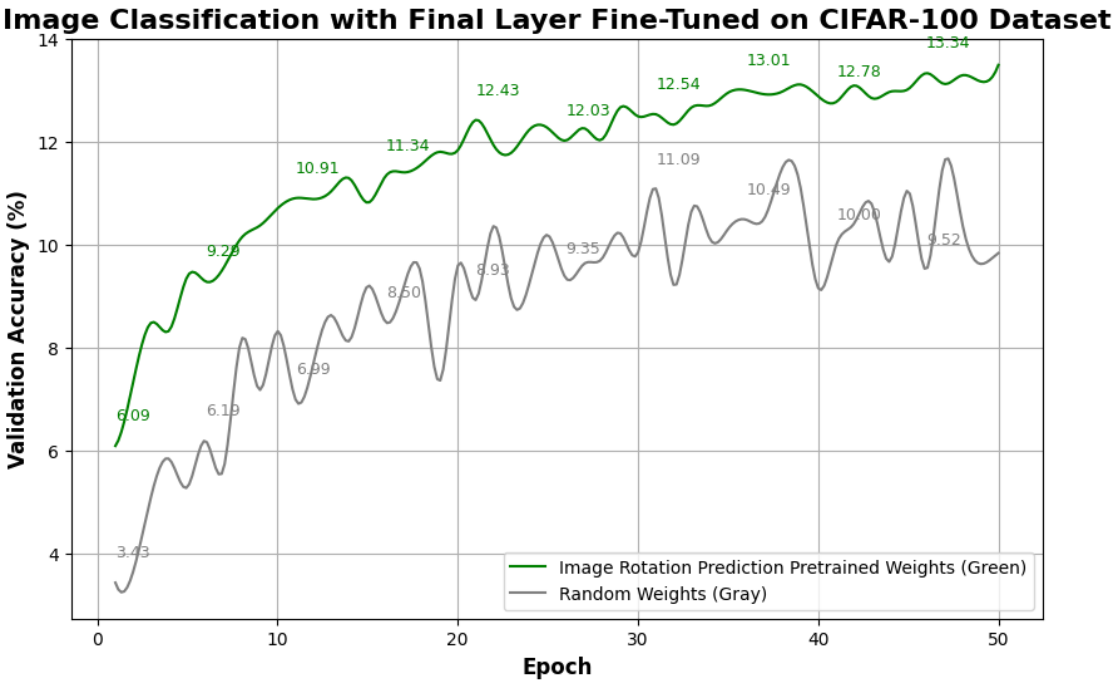
Accuracy Type: Test Accuracy

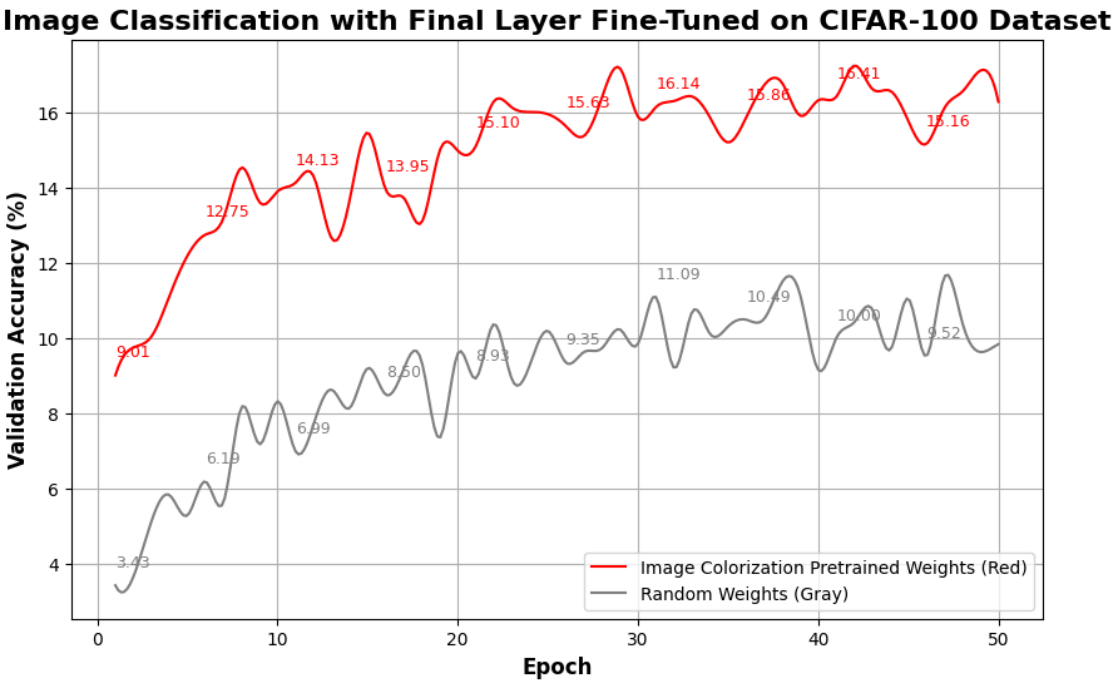
Weights	Test Accuracy
Randomly Initialized	11.70%
Image Rotation Prediction Pretraining	13.16%
Image Colorization Pretraining	18.08%
Image Inpainting Pretraining	28.89%

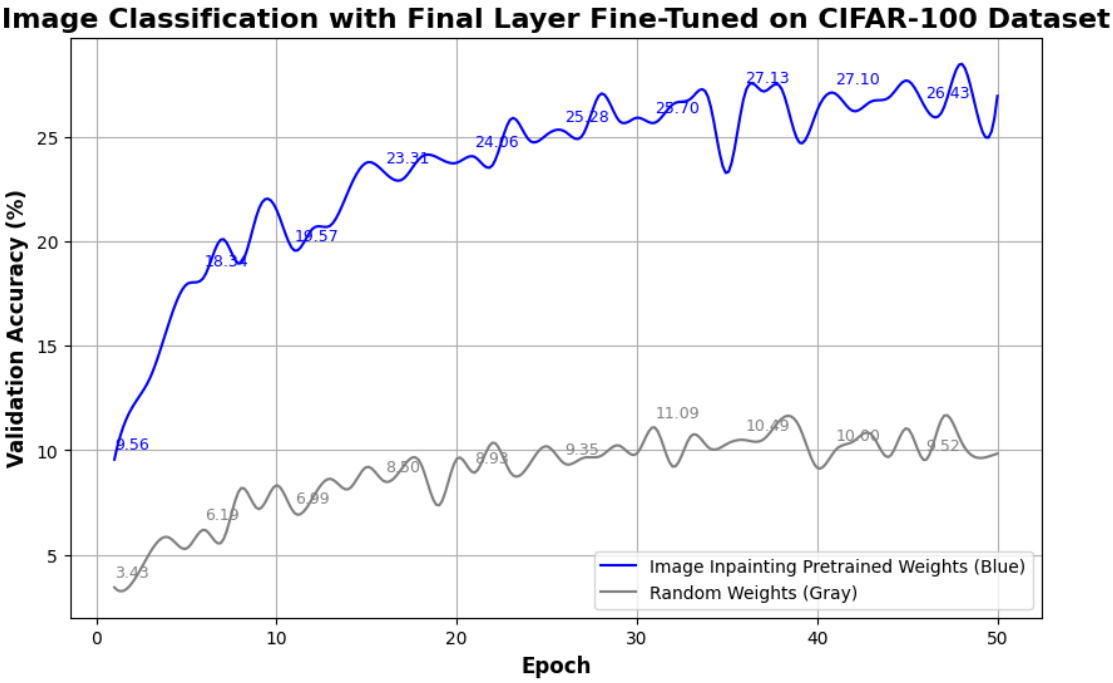


Πίνακας-Διάγραμμα 3.5.2 - 1: Αποτελέσματα ακρίβειας (Test Accuracy) στην ταξινόμηση εικόνων, επανεκπαιδεύοντας μόνο το τελευταίο στρώμα του μοντέλου στη συγκεκριμένη διεργασία, χρησιμοποιώντας τυχαία και προεκπαιδευμένα βάρη από τις τρεις έμμεσες διεργασίες (Image Rotation Prediction, Image Colorization, Image Inpainting) στο σύνολο δεδομένων CIFAR-100.

Αφού παρατέθηκαν οι τελικές τιμές Test Accuracy για το δεύτερο πείραμα, στη συνέχεια θα παρουσιαστούν πιο συγκεκριμένες μετρήσεις για το ίδιο πείραμα, αναλύοντας τις τιμές του Validation Accuracy ανά εποχή. Αυτές οι μετρήσεις θα μας δώσουν μια πιο λεπτομερή εικόνα για την απόδοση του μοντέλου κατά τη διάρκεια της εκπαίδευσης:







Διαγράμματα 3.5.2 - 2,3,4: Αποτελέσματα ακρίβειας (στο validation set) ανά εποχή στην ταξινόμηση εικόνων, επανεκπαιδεύοντας μόνο το τελευταίο στρώμα του μοντέλου στη συγκεκριμένη διεργασία, χρησιμοποιώντας τυχαία βάρη ως σημείο αναφοράς και τα προεκπαιδευμένα βάρη από τις τρεις έμμεσες διεργασίες: Image Rotation Prediction, Image Colorization, Image Inpainting αντίστοιχα, στο σύνολο δεδομένων CIFAR-100.

3.5.3 Αποτελέσματα 3ου πειράματος

Task: Image Classification

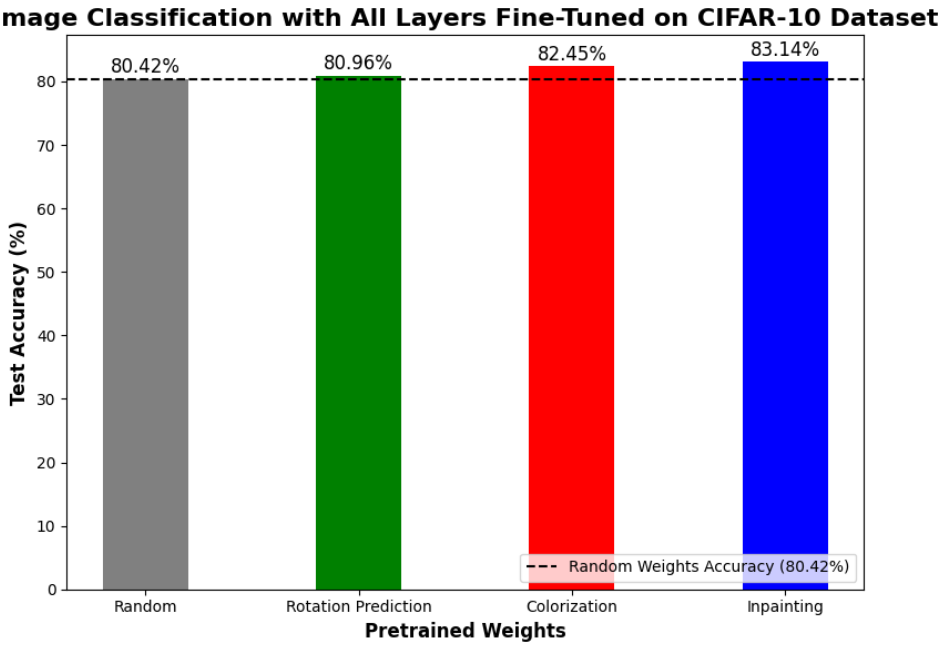
Fine-Tuning: All Layers

Weights: Randomly Initialized, Image Rotation Prediction Pretraining, Image Colorization Pretraining, Image Inpainting Pretraining

Dataset: CIFAR-10

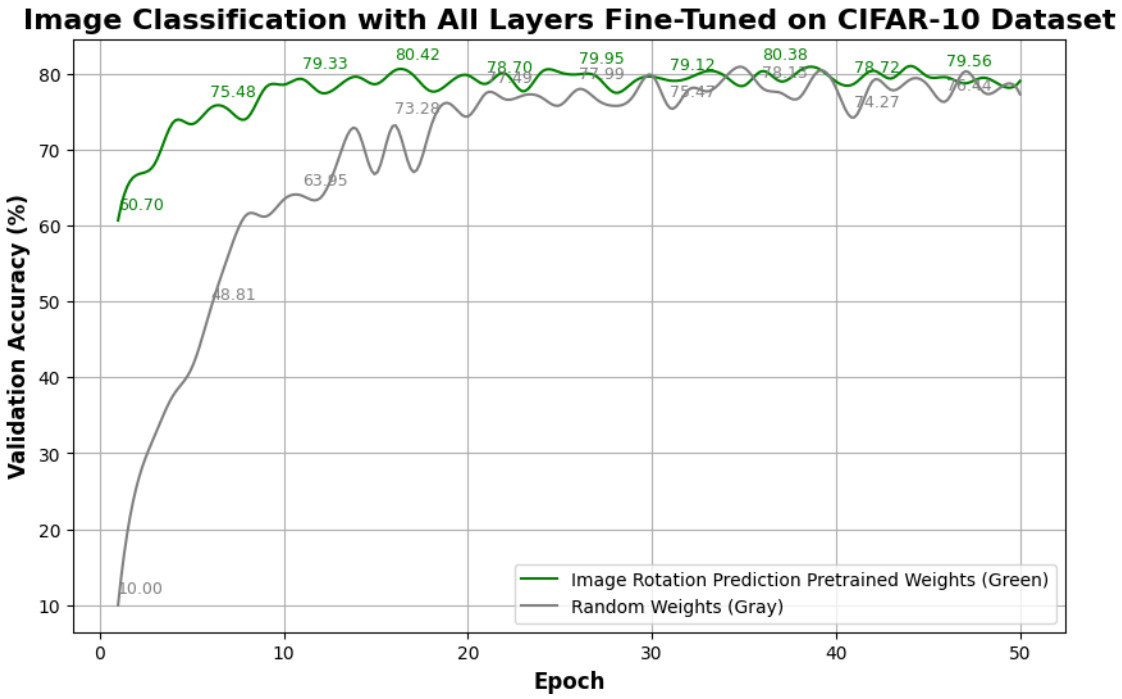
Accuracy Type: Test Accuracy

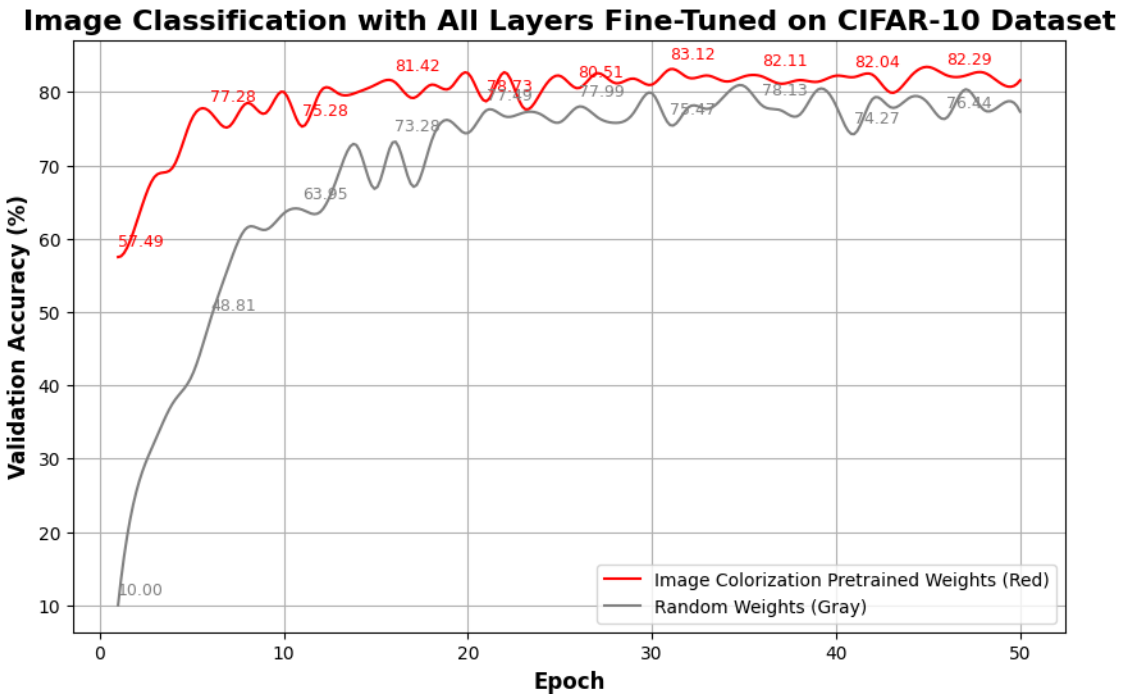
Weights	Test Accuracy
Randomly Initialized	80.42%
Image Rotation Prediction Pretraining	80.96%
Image Colorization Pretraining	82.45%
Image Inpainting Pretraining	83.14%

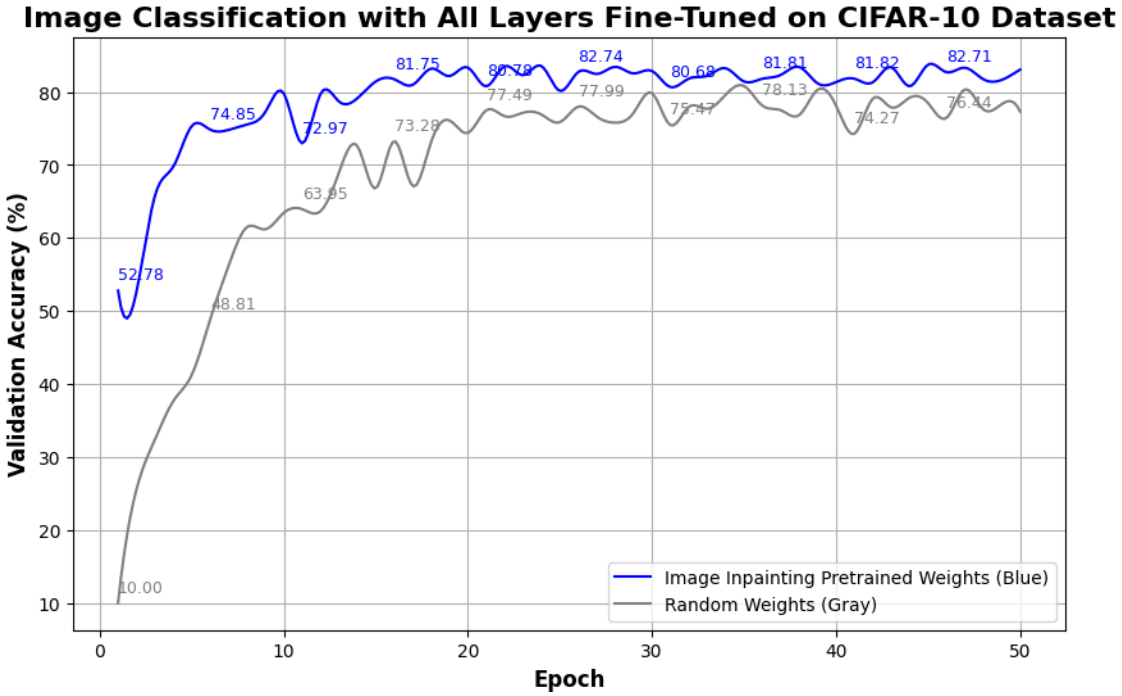


Πίνακας-Διάγραμμα 3.5.3 - 1: Αποτελέσματα ακρίβειας (Test Accuracy) στην ταξινόμηση εικόνων, επανεκπαιδεύοντας όλα τα στρώματα του μοντέλου στη συγκεκριμένη διεργασία, χρησιμοποιώντας τυχαία και προεκπαιδευμένα βάρη από τις τρεις έμμεσες διεργασίες (Image Rotation Prediction, Image Colorization, Image Inpainting) στο σύνολο δεδομένων CIFAR-10.

Αφού παρατέθηκαν οι τελικές τιμές Test Accuracy για το τρίτο πείραμα, στη συνέχεια θα παρουσιαστούν πιο συγκεκριμένες μετρήσεις για το ίδιο πείραμα, αναλύοντας τις τιμές του Validation Accuracy ανά εποχή. Αυτές οι μετρήσεις θα μας δώσουν μια πιο λεπτομερή εικόνα για την απόδοση του μοντέλου κατά τη διάρκεια της εκπαίδευσης:

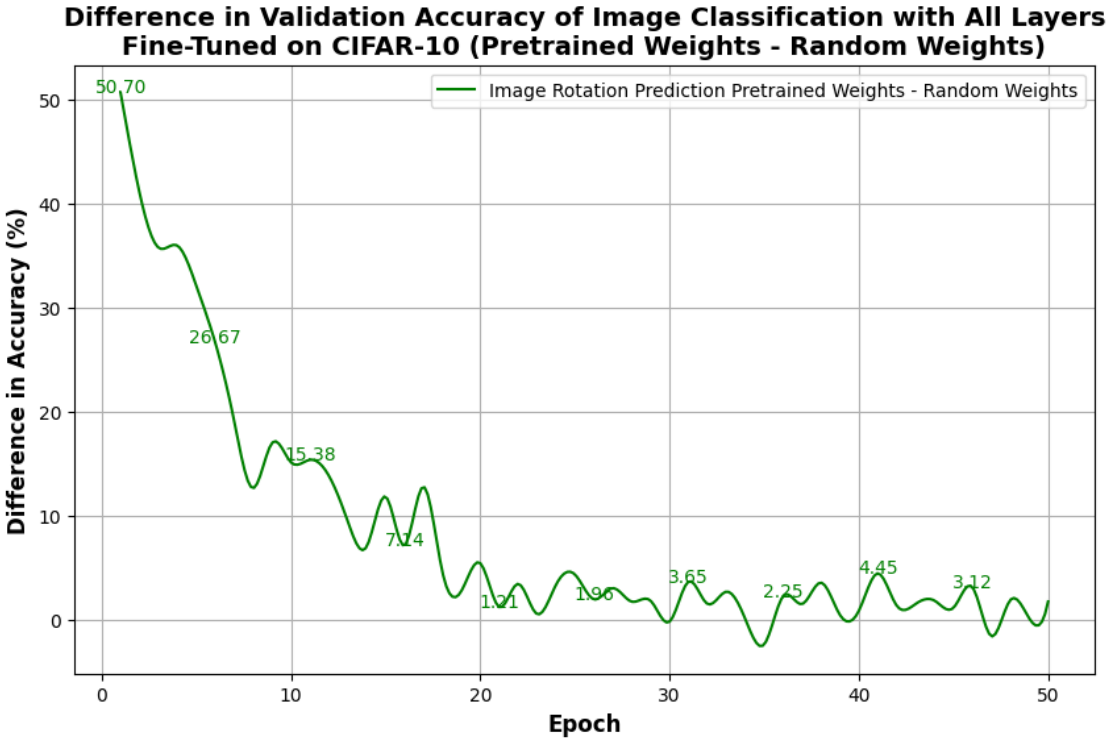


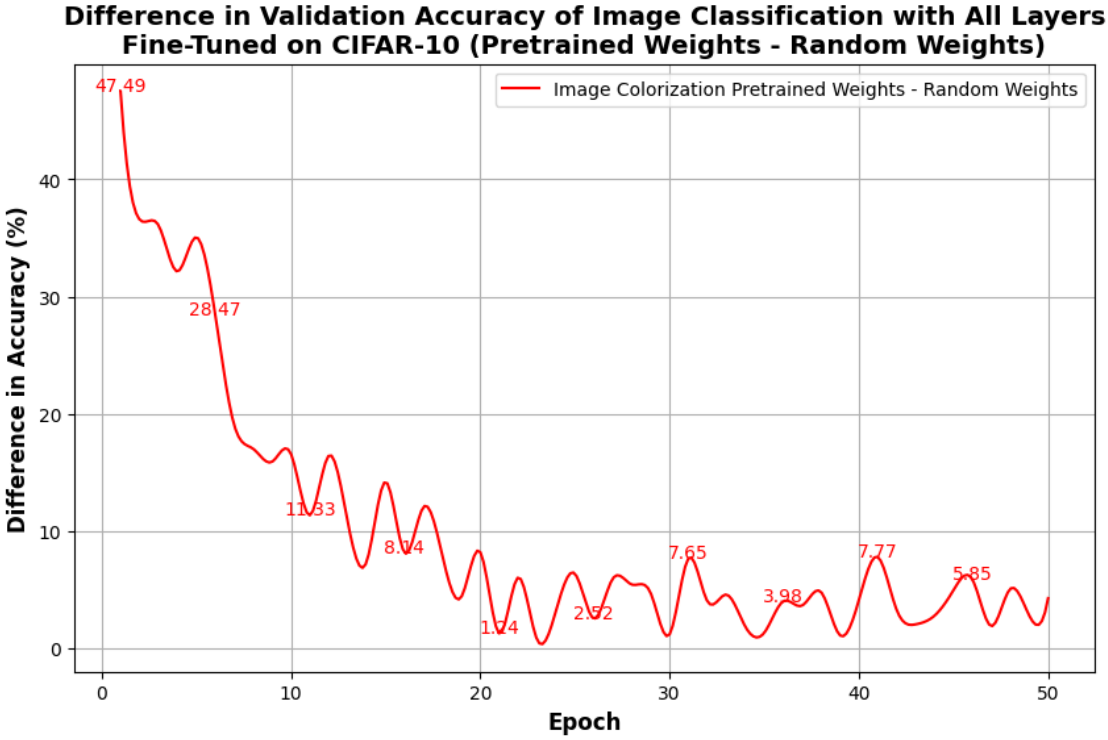


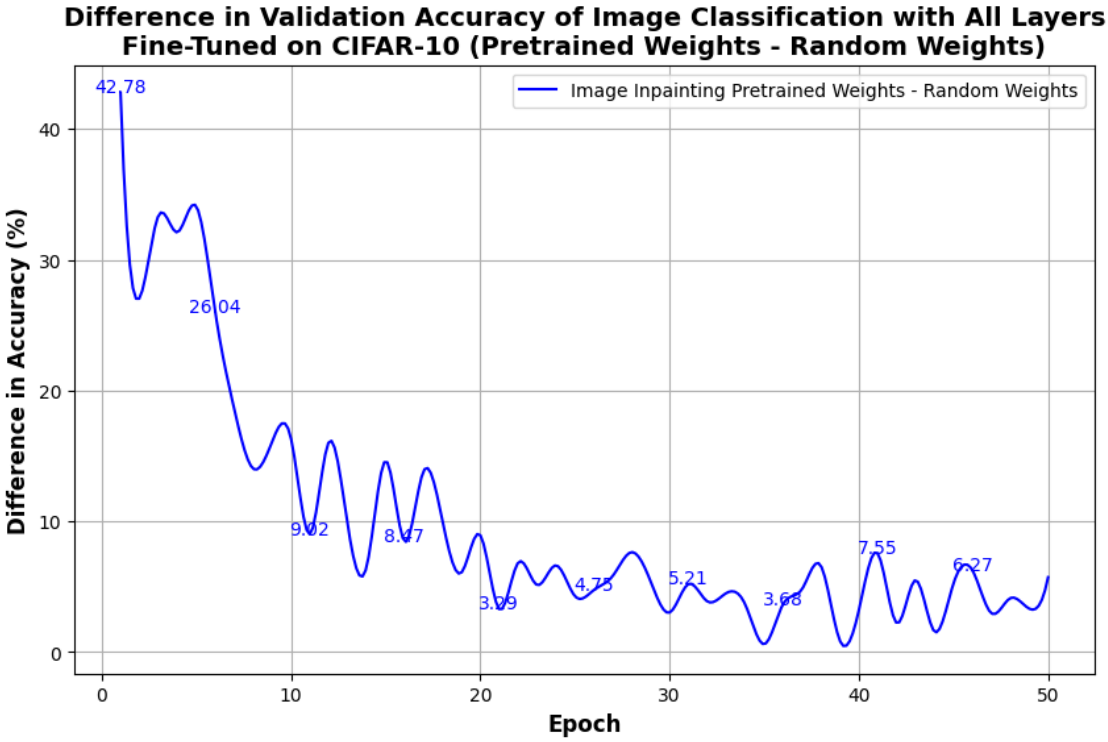


Διαγράμματα 3.5.3 - 2,3,4: Αποτελέσματα ακρίβειας (στο validation set) ανά εποχή στην ταξινόμηση εικόνων, επανεκπαιδεύοντας όλα τα στρώματα του μοντέλου στη συγκεκριμένη διεργασία, χρησιμοποιώντας τυχαία βάρη ως σημείο αναφοράς και τα προεκπαιδευμένα βάρη από τις τρεις έμμεσες διεργασίες: Image Rotation Prediction, Image Colorization, Image Inpainting αντίστοιχα ως αρχικά βάρη, στο σύνολο δεδομένων CIFAR-10.

Επιπλέον, στο συγκεκριμένο όπως και στο επόμενο πείραμα παρουσιάζονται τα διαγράμματα που δείχνουν τη διαφορά της ακρίβειας ανάμεσα στις τιμές που μετρήθηκαν για το validation accuracy του μοντέλου με τα προεκπαιδευμένα βάρη από την κάθε έμμεση διεργασία και τις τιμές που μετρήθηκαν για το validation accuracy του μοντέλου με random weights. Αυτή η σύγκριση θα μας βοηθήσει να κατανοήσουμε καλύτερα την επίδραση των προεκπαιδευμένων βαρών σε σχέση με τα τυχαία βάρη και να εξάγουμε πιο ολοκληρωμένα συμπεράσματα για την απόδοση του μοντέλου ανά τις εποχές, δηλαδή κατά τη διάρκεια της εκπαίδευσης:


Difference in Validation Accuracy of Image Classification with All Layers
Fine-Tuned on CIFAR-10 (Pretrained Weights - Random Weights)
Image Rotation Prediction Pretrained Weights - Random Weights
Difference in Accuracy (%)
Epoch
50.70
26.67
15.38
7.14
1.21
1.96
3.65
2.25
4.45
3.12


Difference in Validation Accuracy of Image Classification with All Layers
Fine-Tuned on CIFAR-10 (Pretrained Weights - Random Weights)
Image Colorization Pretrained Weights - Random Weights
Difference in Accuracy (%)
Epoch
47.49
28.47
11.33
8.14
1.24
2.52
7.65
3.98
7.77
5.85



Διαγράμματα 3.5.3 - 5,6,7: Διαφορά του Validation Accuracy στην ταξινόμηση εικόνων, επανεκπαιδεύοντας όλα τα στρώματα στο συγκεκριμένο task, ανάμεσα στο μοντέλο αρχικοποιημένο με τα προεκπαιδευμένα βάρη και στο μοντέλο αρχικοποιημένο με τυχαία βάρη για κάθε έμμεση διεργασία (Image Rotation Prediction, Image Colorization, Image Inpainting) αντίστοιχα στο σύνολο δεδομένων CIFAR-10.

3.5.4 Αποτελέσματα 4ου πειράματος

Task: Image Classification

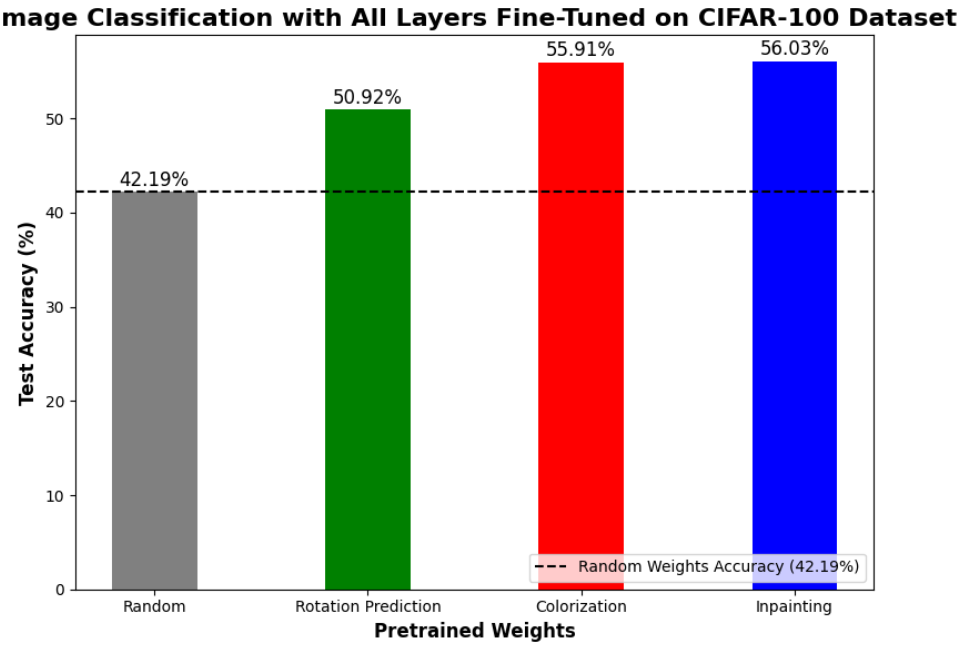
Fine-Tuning: All Layers

Weights: Randomly Initialized, Image Rotation Prediction Pretraining, Image Colorization Pretraining, Image Inpainting Pretraining

Dataset: CIFAR-100

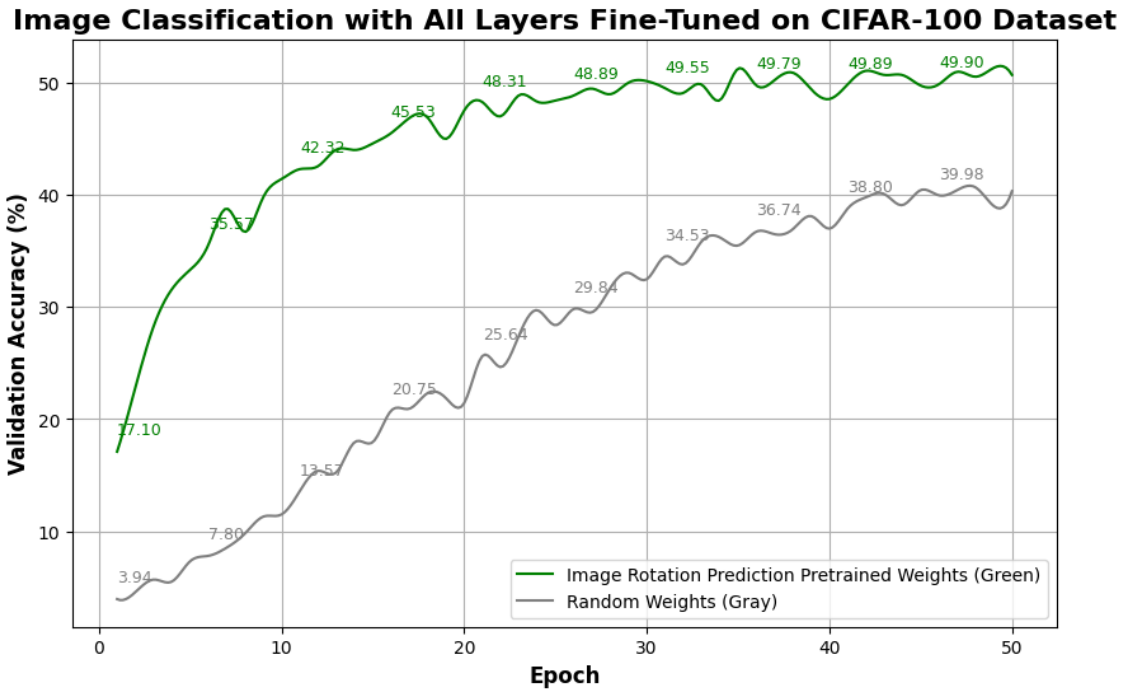
Accuracy Type: Test Accuracy

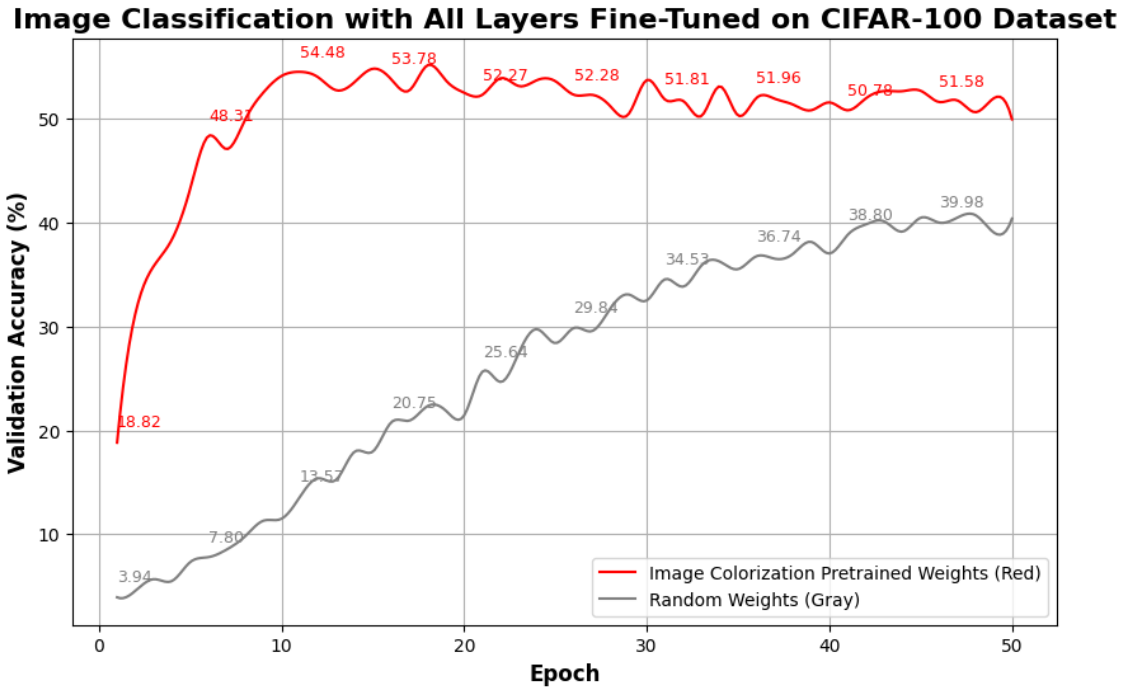
Weights	Test Accuracy
Randomly Initialized	42.19%
Image Rotation Prediction Pretraining	50.92%
Image Colorization Pretraining	55.91%
Image Inpainting Pretraining	56.03%

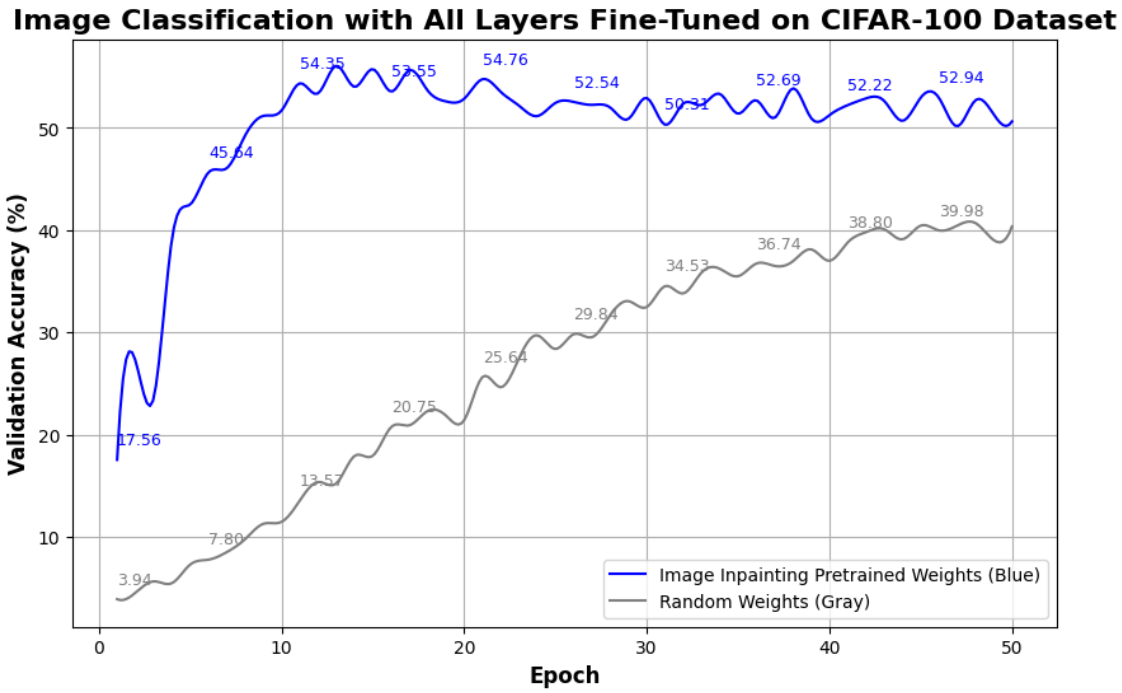


Πίνακας-Διάγραμμα 3.5.4 - 1: Αποτελέσματα ακρίβειας (στο test set) στην ταξινόμηση εικόνων, επανεκπαιδεύοντας όλα τα στρώματα του μοντέλου στη συγκεκριμένη διεργασία, χρησιμοποιώντας τυχαία και προεκπαιδευμένα βάρη από τις τρεις έμμεσες διεργασίες (Image Rotation Prediction, Image Colorization, Image Inpainting) στο σύνολο δεδομένων CIFAR-100.

Αφού παρατέθηκαν οι τελικές τιμές Test Accuracy για το τέταρτο πείραμα, στη συνέχεια θα παρουσιαστούν πιο συγκεκριμένες μετρήσεις για το ίδιο πείραμα, αναλύοντας τις τιμές του Validation Accuracy ανά εποχή. Αυτές οι μετρήσεις θα μας δώσουν μια πιο λεπτομερή εικόνα για την απόδοση του μοντέλου κατά τη διάρκεια της εκπαίδευσης:

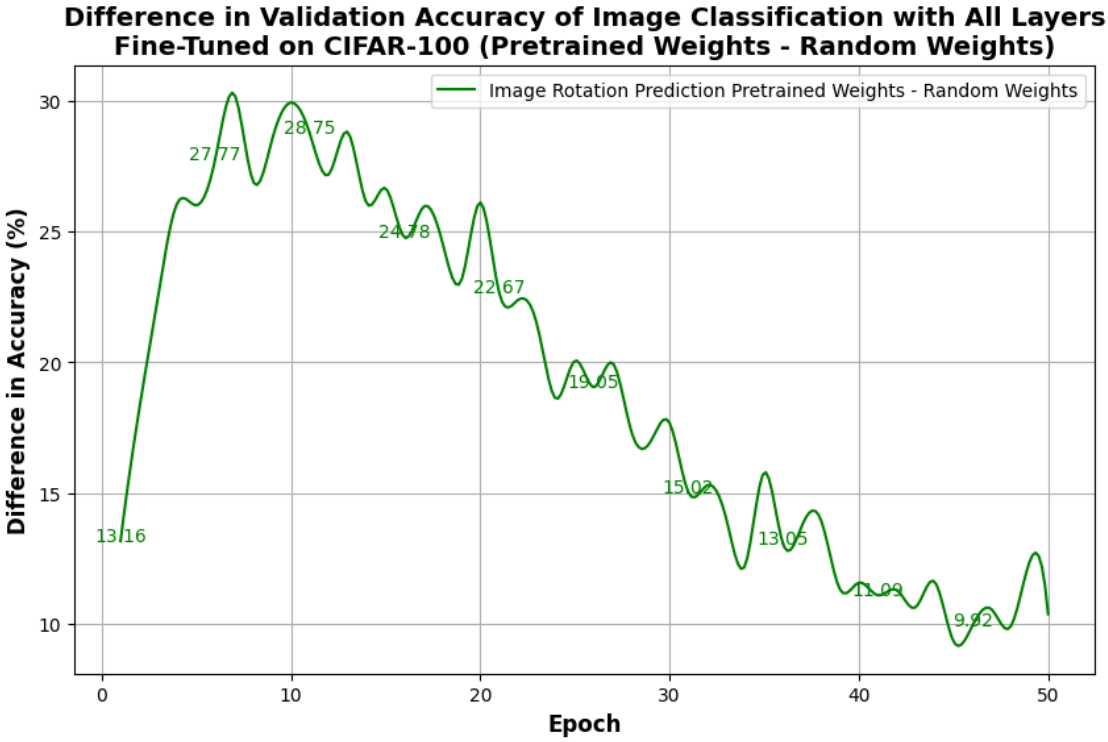


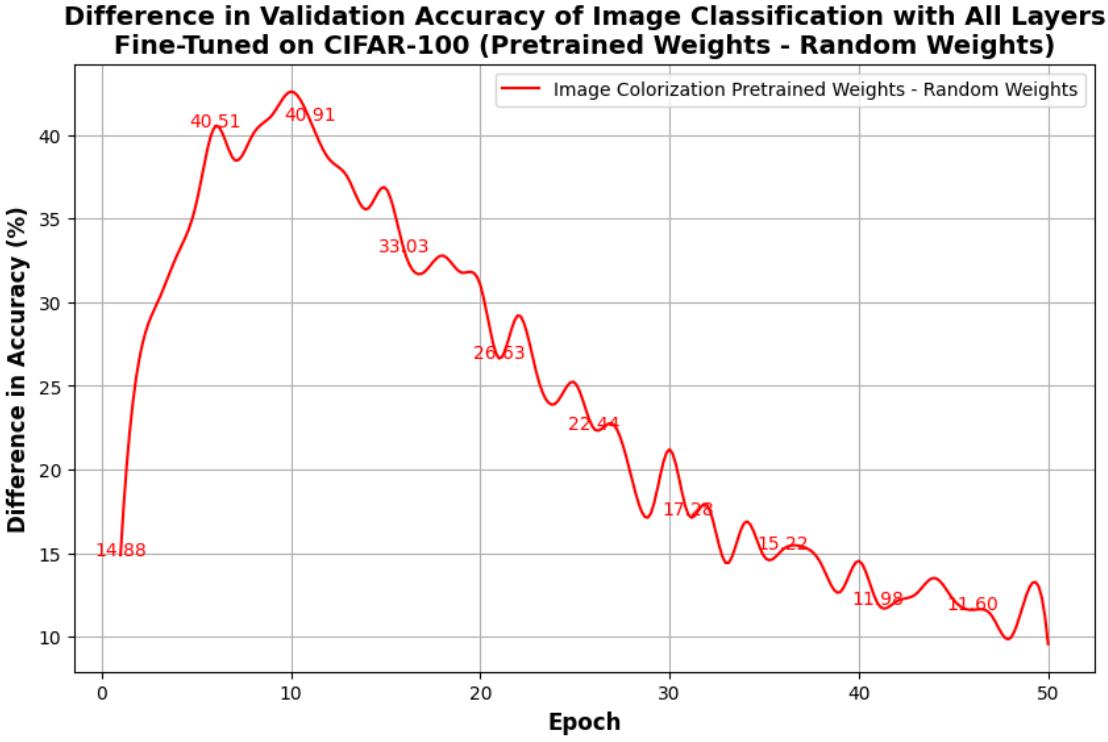


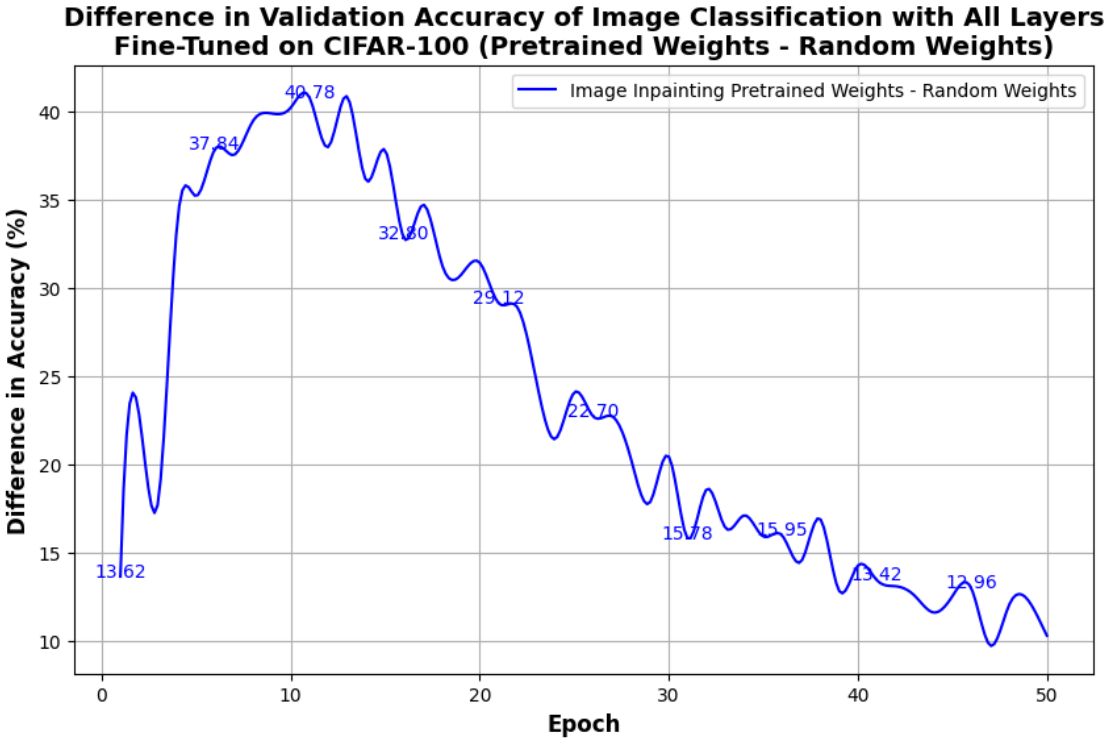


Διαγράμματα 3.5.4 - 2,3,4: Αποτελέσματα ακρίβειας (στο validation set) ανά εποχή στην ταξινόμηση εικόνων, επανεκπαιδεύοντας όλα τα στρώματα του μοντέλου στη συγκεκριμένη διεργασία, χρησιμοποιώντας τυχαία βάρη ως σημείο αναφοράς και τα προεκπαιδευμένα βάρη από τις τρεις έμμεσες διεργασίες: Image Rotation Prediction, Image Colorization, Image Inpainting αντίστοιχα ως αρχικά βάρη, στο σύνολο δεδομένων CIFAR-100.

Επιπλέον, παρουσιάζονται τα διαγράμματα που δείχνουν τη διαφορά της ακρίβειας ανάμεσα στις τιμές που μετρήθηκαν για το validation accuracy του μοντέλου με τα προεκπαιδευμένα βάρη από την κάθε έμμεση διεργασία και τις τιμές που μετρήθηκαν για το validation accuracy του μοντέλου με random weights. Αυτή η σύγκριση θα μας βοηθήσει να κατανοήσουμε καλύτερα την επίδραση των προεκπαιδευμένων βαρών σε σχέση με τα τυχαία βάρη και να εξάγουμε πιο ολοκληρωμένα συμπεράσματα για την απόδοση του μοντέλου ανά τις εποχές, δηλαδή κατά τη διάρκεια της εκπαίδευσης:


Difference in Validation Accuracy of Image Classification with All Layers
Fine-Tuned on CIFAR-100 (Pretrained Weights - Random Weights)
Image Rotation Prediction Pretrained Weights - Random Weights
Difference in Accuracy (%)
Epoch
13.16
27.77
28.75
24.78
22.67
19.05
15.02
13.05
11.09
9.92

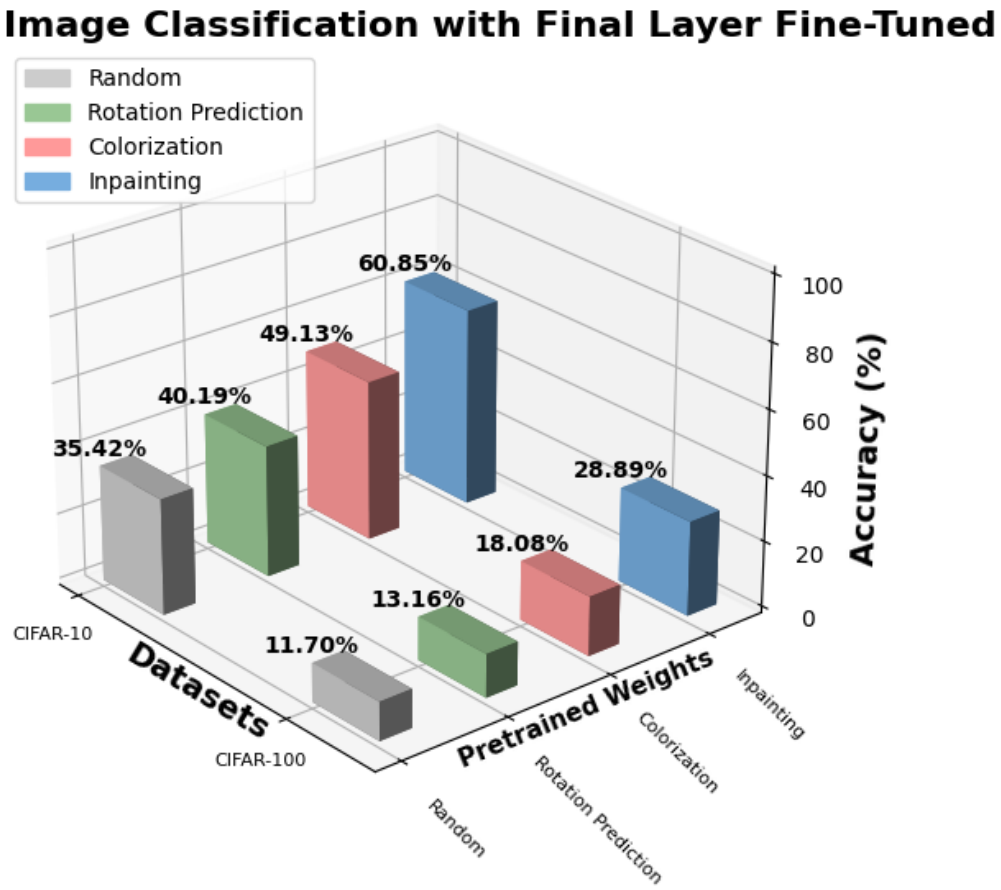

Difference in Validation Accuracy of Image Classification with All Layers
Fine-Tuned on CIFAR-100 (Pretrained Weights - Random Weights)
Image Colorization Pretrained Weights - Random Weights
Difference in Accuracy (%)
Epoch
14.88
40.51
40.91
33.03
26.63
22.44
17.28
15.22
11.98
11.60



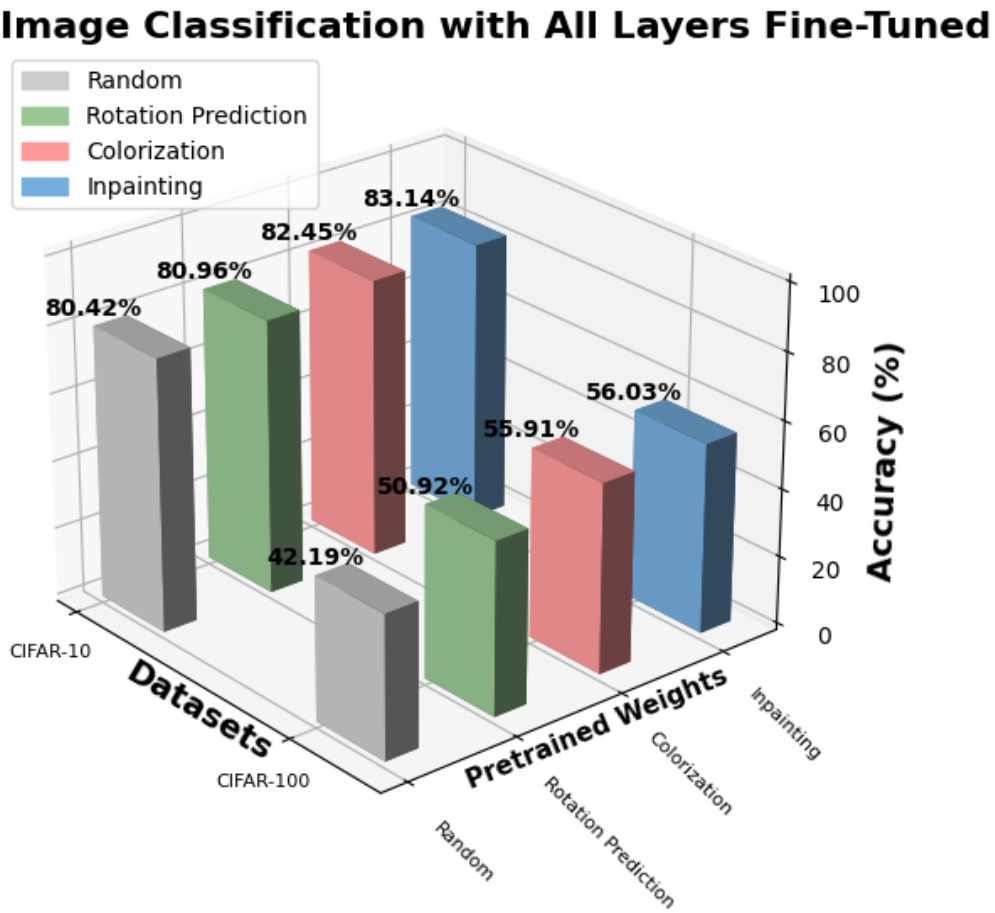
Διαγράμματα 3.5.4 - 5,6,7: Διαφορά του Validation Accuracy στην ταξινόμηση εικόνων, επανεκπαιδεύοντας όλα τα στρώματα στο συγκεκριμένο task, ανάμεσα στο μοντέλο αρχικοποιημένο με τα προεκπαιδευμένα βάρη και στο μοντέλο αρχικοποιημένο με τυχαία βάρη για κάθε έμμεση διεργασία (Image Rotation Prediction, Image Colorization, Image Inpainting) αντίστοιχα στο σύνολο δεδομένων CIFAR-100.

3.5.5 Συγκεντρωτική Απεικόνιση Αποτελεσμάτων

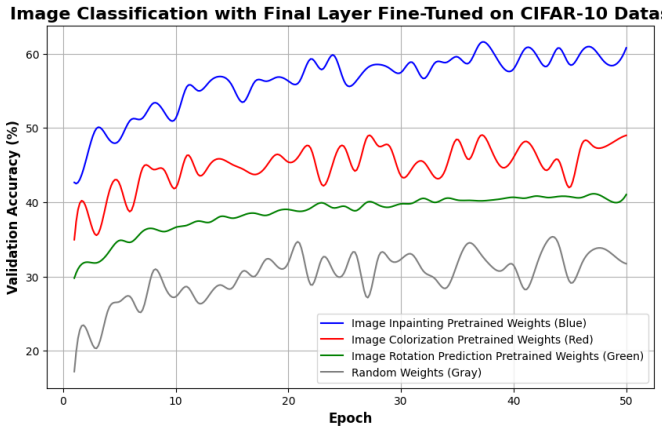
Τέλος, θα συνοψίσουμε όλες τις μετρήσεις στα παρακάτω διαγράμματα, ώστε να διευκολύνουμε την κατανόηση των διαφορών ανάμεσα στα tasks και να παρέχουμε μια πιο ολοκληρωμένη εικόνα της απόδοσης του μοντέλου. Με τη συγκεντρωτική απεικόνιση των αποτελεσμάτων, θα καταστεί ευκολότερη η εξαγωγή συμπερασμάτων σχετικά με την επίδραση των προεκπαιδευμένων βαρών της κάθε έμμεσης διεργασίας, καθώς και η σύγκρισή τους με τα τυχαία βάρη.

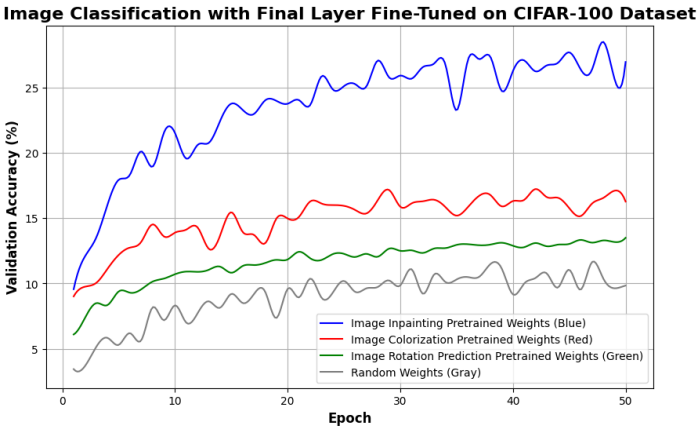


Διαγράμματα 3.5.5 - 1: Αποτελέσματα ακρίβειας (Test Accuracy) στην ταξινόμηση εικόνων, επανεκπαιδεύοντας μόνο το τελευταίο στρώμα του μοντέλου στη συγκεκριμένη διεργασία, χρησιμοποιώντας τυχαία και προεκπαιδευμένα βάρη από τις τρεις έμμεσες διεργασίες (Image Rotation Prediction, Image Colorization, Image Inpainting) στα σύνολα δεδομένων CIFAR-10 και CIFAR-100.

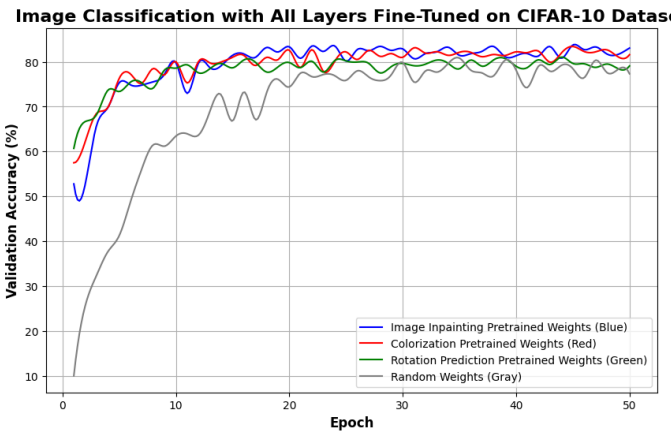


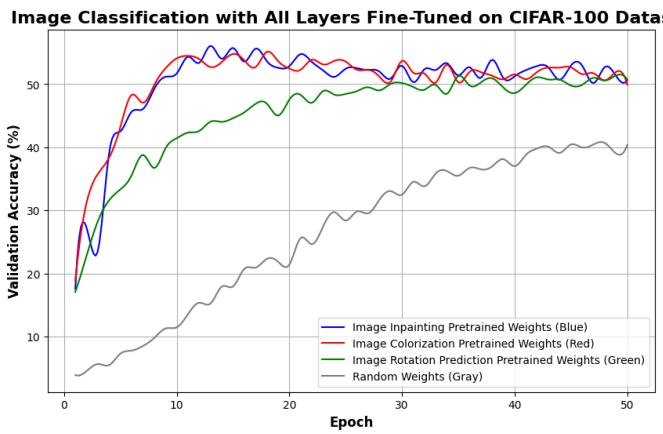
Διάγραμμα 3.5.5 - 2: Αποτελέσματα ακρίβειας (Test Accuracy) στην ταξινόμηση εικόνων, επανεκπαιδεύοντας όλα τα στρώματα του μοντέλου στη συγκεκριμένη διεργασία, χρησιμοποιώντας τυχαία και προεκπαιδευμένα βάρη από τις τρεις έμμεσες διεργασίες (Image Rotation Prediction, Image Colorization, Image Inpainting) στα σύνολα δεδομένων CIFAR-10 και CIFAR-100.





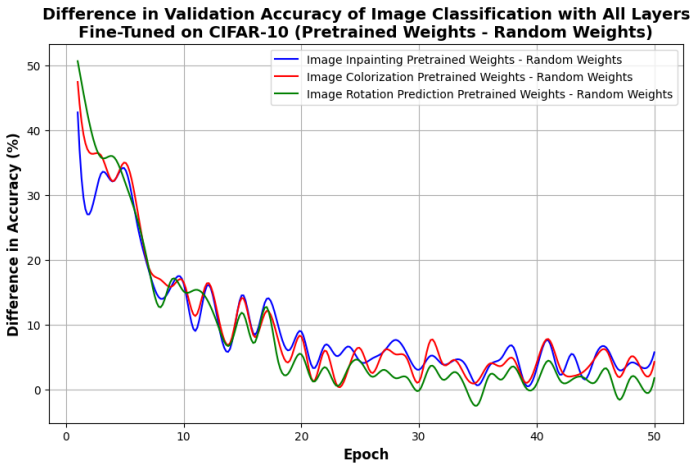
Διαγράμματα 3.5.5 – 3,4: Αποτελέσματα ακρίβειας (στο validation set) ανά εποχή στην ταξινόμηση εικόνων, επανεκπαιδεύοντας μόνο το τελευταίο στρώμα του μοντέλου στη συγκεκριμένη διεργασία, χρησιμοποιώντας τυχαία βάρη ως σημείο αναφοράς και τα προεκπαιδευμένα βάρη από τις τρεις έμμεσες διεργασίες: Image Rotation Prediction, Image Colorization, Image Inpainting ως αρχικά βάρη, στα σύνολα δεδομένων CIFAR-10 και CIFAR-100.

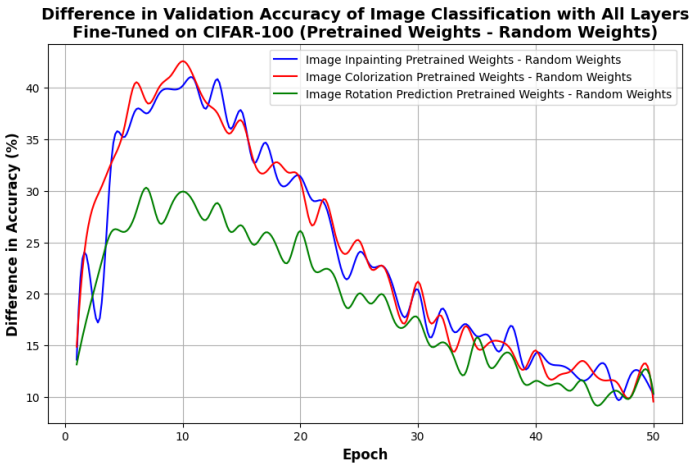




Δ

ιαγράμματα 3.5.5 – 5,6: Αποτελέσματα ακρίβειας (στο validation set) ανά εποχή στην ταξινόμηση εικόνων, επανεκπαιδεύοντας όλα τα στρώματα του μοντέλου στη συγκεκριμένη διεργασία, χρησιμοποιώντας τυχαία βάρη ως σημείο αναφοράς και τα προεκπαιδευμένα βάρη από τις τρεις έμμεσες διεργασίες: Image Rotation Prediction, Image Colorization, Image Inpainting ως αρχικά βάρη, στα σύνολα δεδομένων CIFAR-10 και CIFAR-100.





Διαγράμματα 3.5.5 - 7,8: Διαφορά του Validation Accuracy στην ταξινόμηση εικόνων, επανεκπαιδεύοντας όλα τα στρώματα στο συγκεκριμένο task, ανάμεσα στο μοντέλο αρχικοποιημένο με τα προεκπαιδευμένα βάρη και στο μοντέλο αρχικοποιημένο με τυχαία βάρη για τις τρεις έμμεσες διεργασίες (Image Rotation Prediction, Image Colorization, Image Inpainting) στα σύνολα δεδομένων CIFAR-10 και CIFAR-100.

3.6 Σχολιασμός Αποτελεσμάτων - Συμπεράσματα

Επιδόσεις με τη χρήση Έμμεσων Διεργασιών για την προ-εκπαίδευση των βαρών σε σχέση με τα τυχαία βάρη

Σύμφωνα με τις μετρήσεις και όπως φαίνεται στα διαγράμματα που παρατίθενται στην προηγούμενη ενότητα, οι επιδόσεις όλων των proxy tasks ήταν σταθερά (σε όλα τα πειράματα) καλύτερες από εκείνες που επιτεύχθηκαν με τη χρήση random weights. Αυτό επιβεβαιώνει την αρχική μας προσδοκία ότι η *SSL* προσφέρει σημαντικό πλεονέκτημα, καθώς τα προεκπαιδευμένα βάρη έχουν ήδη μάθει βασικά χαρακτηριστικά από τα proxy tasks. Σε αντίθεση, τα τυχαία βάρη δεν φέρουν καμία αρχική πληροφορία και απαιτούν περισσότερο χρόνο και δεδομένα για να μάθουν ουσιαστικά χαρακτηριστικά. Επομένως, τα proxy tasks προσέφεραν σημαντική βελτίωση στις επιδόσεις σε όλες τις μετρήσεις, επιβεβαιώνοντας τον σκοπό της αυτο-εποπτευόμενης μάθησης.

Επιδόσεις στα Διαφορετικά Datasets (CIFAR-10, CIFAR-100)

Σύμφωνα με τα αποτελέσματα, οι επιδόσεις στο CIFAR-100 ήταν χαμηλότερες σε σύγκριση με το CIFAR-10, κάτι που επιβεβαιώνει την αρχική μας υπόθεση ότι το task ταξινόμησης στο CIFAR-100 είναι πιο πολύπλοκο λόγω του μεγαλύτερου αριθμού κλάσεων (100 έναντι 10). Αυτό είναι εμφανές τόσο στην περίπτωση του fine-tuning μόνο του τελευταίου επιπέδου όσο και στην περίπτωση του fine-tuning ολόκληρου του μοντέλου, όπου οι επιδόσεις στο CIFAR-100 παρέμειναν χαμηλότερες.

Ωστόσο, τα pretrained weights που εκπαιδεύτηκαν στο CIFAR-100 φαίνεται να μεταφέρουν ικανοποιητικά τις γνώσεις τους στο CIFAR-10. Παρά την διαφορετικότητα των συγκεκριμένων συνόλων δεδομένων, τα προεκπαιδευμένα βάρη που προέκυψαν από το CIFAR-100 συνέβαλαν στην επίτευξη καλών επιδόσεων στο CIFAR-10, όπως αναμενόταν. Παρά το γεγονός ότι τα proxy tasks εκπαιδεύτηκαν σε 90 επιπλέον κλάσεις που δεν υπάρχουν στο CIFAR-10 και θα μπορούσαν να θεωρηθούν άχρηστες, το μοντέλο κατάφερε να μάθει σημαντικές πληροφορίες που ήταν χρήσιμες και στο CIFAR-10, επιτυγχάνοντας ικανοποιητικές αποδόσεις.

Επιπλέον, αυτό μας δείχνει ότι τα proxy tasks μαθαίνουν βασικά χαρακτηριστικά που σχετίζονται με τον προσανατολισμό, το χρώμα και τη συνοχή της εικόνας (για τα τρία proxy tasks αντίστοιχα) και όχι με βάση την κλάση των εικόνων. Αυτό επιτρέπει στα μοντέλα να μεταφέρουν χρήσιμες πληροφορίες ανεξαρτήτως των κλάσεων στις οποίες ανήκουν οι εικόνες. Αυτό συμβαίνει και επειδή χρησιμοποιήσαμε το CIFAR-100 ως unlabeled dataset για την εκπαίδευση στις έμμεσες διεργασίες. Επομένως, ο ισχυρισμός ότι το μοντέλο έμαθε από τα labels, ακόμα και αν δεν τα χρησιμοποιήσαμε άμεσα, δεν ευσταθεί διότι τα αποτελέσματα επιβεβαιώνουν ότι το μοντέλο μαθαίνει βάσει χαρακτηριστικών όπως ο προσανατολισμός, το χρώμα και η συνοχή της εικόνας, και όχι βάσει των κλάσεων των εικόνων.

Συμπερασματικά, οι επιδόσεις ήταν χαμηλότερες στο CIFAR-100 λόγω της πολυπλοκότητας του task, αλλά τα proxy tasks απέδειξαν ότι μπορούν να μεταφέρουν χρήσιμη πληροφορία στο CIFAR-10, διατηρώντας ικανοποιητικές αποδόσεις.

Σύγκριση πληροφορίας ανάμεσα στα προ-εκπαιδευμένα βάρη από τις έμμεσες διεργασίες και τα τυχαία βάρη, με fine-tuning μόνο στο τελευταίο επίπεδο του μοντέλου

Στο CIFAR-10, παρατηρούμε μια αύξηση απόδοσης σε σχέση με την αρχική επίδοση των τυχαίων βαρών:

- 13.47% για το rotation (από 35.42% σε 40.19%),
- 38.71% για το colorization (από 35.42% σε 49.13%),
- 71.80% για το inpainting (από 35.42% σε 60.85%).

Αντίστοιχα, στο CIFAR-100 παρατηρούμε μια αύξηση απόδοσης σε σχέση με τα τυχαία βάρη:

- 12.48% για το rotation (από 11.70% σε 13.16%),
- 54.53% για το colorization (από 11.70% σε 18.08%),
- 146.92% για το inpainting (από 11.70% σε 28.89%).

Επομένως, με αυτήν την άμεση σύγκριση των επιδόσεων, μπορούμε εύκολα να αντιληφθούμε ότι η πληροφορία που έμαθαν τα βάρη στα proxy tasks είναι ιδιαίτερα χρήσιμη και βοηθά σημαντικά στο task της ταξινόμησης εικόνων.

Σύγκριση τελικών επιδόσεων ανάμεσα στα προ-εκπαιδευμένα βάρη από τις έμμεσες διεργασίες και τα τυχαία βάρη, με fine-tuning σε όλα τα επίπεδα του μοντέλου

Στο CIFAR-10, παρατηρούμε αύξηση σε σχέση με την αρχική απόδοση των τυχαίων βαρών:

- 0.67% για το rotation (από 80.42% σε 80.96%),
- 2.52% για το colorization (από 80.42% σε 82.45%),
- 3.38% για το inpainting (από 80.42% σε 83.14%).

Στο CIFAR-100, παρατηρούμε αύξηση:

- 20.69% για το rotation (από 42.19% σε 50.92%),
- 32.52% για το colorization (από 42.19% σε 55.91%),
- 32.80% για το inpainting (από 42.19% σε 56.03%).

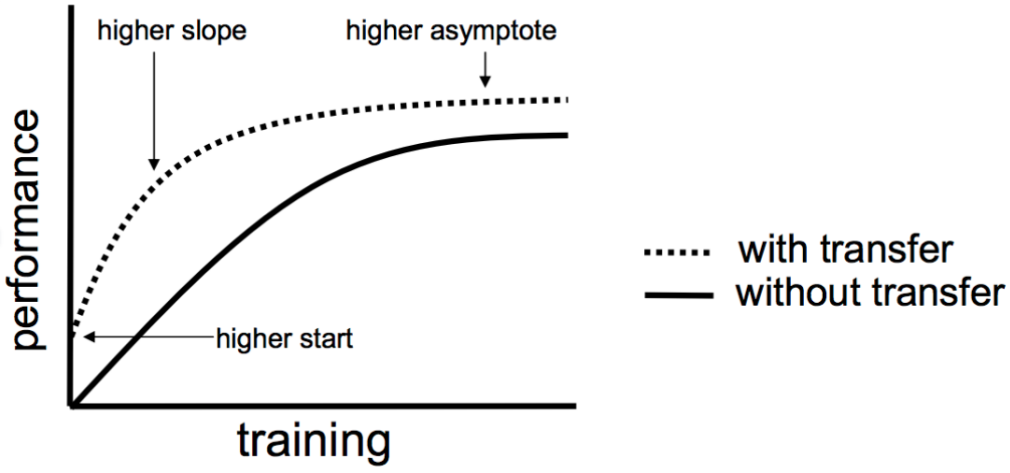
Σύμφωνα με τα αποτελέσματα, βλέπουμε ότι με το fine-tuning σε όλα τα επίπεδα του μοντέλου, οι τελικές επιδόσεις των proxy tasks είναι ελάχιστα καλύτερες από αυτές των random weights. Αυτό συμβαίνει διότι τα βάρη, είτε είναι τυχαία είτε προεκπαιδευμένα, χρησιμοποιούνται μόνο για αρχικοποίηση και στη συνέχεια αλλάζουν κατά τη διάρκεια της εκπαίδευσης. Παρ' όλα αυτά, τα pre-trained weights βοηθούν το μοντέλο να συγκλίνει ελαφρώς καλύτερα.

Ταχύτητα επίτευξης ικανοποιητικής επίδοσης με fine-tuning σε όλα τα επίπεδα του μοντέλου

Σύμφωνα με τα αποτελέσματα, είναι εμφανές στα διαγράμματα που απεικονίζουν την ακρίβεια ανά εποχή ότι η ταχύτητα με την οποία τα proxy tasks επιτρέπουν στο μοντέλο να φτάσει σε υψηλά επίπεδα απόδοσης είναι πολύ μεγαλύτερη σε σχέση με τα τυχαία βάρη. Συγκεκριμένα, στο CIFAR-10, με τα τρία proxy tasks (rotation prediction, colorization, inpainting) έχουμε επιτύχει accuracy 75% στις πρώτες 5 εποχές (epochs) και με τα τρία pretrained weights, ενώ φτάνουμε στο 80% στις πρώτες 10 εποχές. Μετά από εκεί, το καθένα συγκλίνει αντίστοιχα στις τελικές τιμές που αναφέραμε παραπάνω. Αντίθετα, με τα random weights, στην 5η εποχή το accuracy είναι περίπου 40-45% και στις 10 εποχές φτάνει περίπου στο 61%.

Αντίστοιχα, στο CIFAR-100, παρατηρούμε ότι με το rotation πετυχαίνουμε 41% στα πρώτα 10 εποχές, ενώ με το colorization και το inpainting η μέγιστη απόδοση φτάνει στο 55% ήδη από τις πρώτες 10 εποχές. Τα random weights, ωστόσο, παραμένουν στο 11% κατά τις πρώτες 10 εποχές.

Οι καμπύλες που βλέπουμε στα διαγράμματα ταιριάζουν απόλυτα με την παρακάτω εικόνα, η οποία παρουσιάζεται στο paper "An analysis of transfer learning for domain mismatched text-independent speaker verification" από τους Chunlei Zhang, Shivesh Ranjan, John H.L. Hansen [54]. Η εικόνα δείχνει ότι, γενικότερα στο transfer learning, στην αρχή της εκπαίδευσης έχουμε μια υψηλότερη αρχική τιμή (higher start), στις πρώτες εποχές παρατηρείται μια μεγαλύτερη κλίση (higher slope), και προς τα τέλη της εκπαίδευσης παρατηρείται μια υψηλότερη ασύμπτωτη τιμή (higher asymptote). Αυτή η ανάλυση επιβεβαιώνει τη λογική των μετρήσεών μας, η οποία αποτυπώνεται ξεκάθαρα στα διαγράμματα, δείχνοντας ότι τα pretrained weights προσφέρουν ταχύτερη και υψηλότερη επίδοση σε σχέση με τα τυχαία βάρη.



Εικόνα 3.6 - 1: Αναπαράσταση της διαφοράς μεταξύ transfer learning και εκπαίδευσης από τυχαία αρχικοποίηση.
(Πηγή: Zhang, C., Ranjan, S., & Hansen, J. H. (2018, June). An Analysis of Transfer Learning for Domain Mismatched Text-independent Speaker Verification. In *Odyssey* (pp. 181-186).)

Αυτή η διαφορά γίνεται ακόμη πιο εμφανής στα διαγράμματα που δείχνουν τη διαφορά στην ακρίβεια ανάμεσα σε κάθε proxy task και τα random weights, αναδεικνύοντας την ταχύτερη επίτευξη υψηλών επιδόσεων που επιτυγχάνεται με τη χρήση των pretrained weights.

Έμμεση διεργασία με τις καλύτερες επιδόσεις

Όλα τα pretrained weights από τα αντίστοιχα proxy tasks παρουσίασαν καλύτερες επιδόσεις σε σχέση με τα random weights. Η σειρά από το καλύτερο προς το χειρότερο ήταν: 1) Image Inpainting, 2) Image Colorization, 3) Image Rotation Prediction.

Αυτό μας δείχνει ότι το Image Inpainting ήταν το πιο αποδοτικό task, πιθανώς επειδή είναι μια σύνθετη διεργασία που απαιτεί από το μοντέλο να μάθει πώς να συμπληρώνει κενά στις εικόνες, διατηρώντας τη συνοχή τους. Αυτή η διαδικασία αναγκάζει το δίκτυο να κατανοήσει τόσο το περιεχόμενο της εικόνας όσο και το ευρύτερο πλαίσιο (context) της, ενισχύοντας έτσι την ικανότητά του να αναπαριστά πολύπλοκες σχέσεις και χαρακτηριστικά.

Από την άλλη, το Image Rotation Prediction αποδείχθηκε πιο "αδύναμο", καθώς είναι μια πιο απλή διεργασία που εστιάζει μόνο στον προσανατολισμό των αντικειμένων. Αν και παρέχει κάποια χρήσιμη πληροφορία στο μοντέλο, δεν είναι τόσο πλούσιο σε χαρακτηριστικά όσο τα άλλα δύο tasks, με αποτέλεσμα η πληροφορία που μεταφέρει να είναι λιγότερο χρήσιμη στο downstream task, όπως φαίνεται από τις ελαφρώς αυξημένες, αλλά περιορισμένες, επιδόσεις του.

3.7 Επίλογος και Μελλοντικές Επεκτάσεις

Συνοψίζοντας τα αποτελέσματα αυτής της διπλωματικής εργασίας, καταλήγουμε στο συμπέρασμα ότι η Αυτο-Εποπτευόμενη Μάθηση μέσω της χρήσης έμμεσων διεργασιών προσφέρει σημαντικά πλεονεκτήματα στην εκπαίδευση νευρωνικών δικτύων για ταξινόμηση εικόνων. Συγκεκριμένα, τα προεκπαιδευμένα βάρη από τα proxy tasks αποδείχθηκαν πιο αποδοτικά σε σχέση με τα τυχαία βάρη, τόσο όσον αφορά τις τελικές επιδόσεις όσο και την ταχύτητα εκπαίδευσης. Η διεργασία Image Inpainting αναδείχθηκε ως η πιο αποτελεσματική, ακολουθούμενη από το Image Colorization, ενώ το Image Rotation Prediction παρουσίασε τα χαμηλότερα αποτελέσματα, αν και παραμένει καλύτερο από την εκπαίδευση με τυχαία βάρη.

Η προσέγγιση αυτή προσφέρει σημαντικά πλεονεκτήματα, καθώς επιτρέπει την εκμετάλλευση μη επισημασμένων δεδομένων, μειώνοντας τις ανάγκες για εξειδικευμένο labeling και επιταχύνοντας τη διαδικασία εκπαίδευσης. Το γεγονός ότι τα προεκπαιδευμένα βάρη επέτρεψαν στο μοντέλο να συγκλίνει ταχύτερα και να πετύχει υψηλότερες επιδόσεις με λιγότερους πόρους υπογραμμίζει την αξία της μεταφοράς μάθησης σε προβλήματα που δεν έχουν πολλά επισημασμένα δεδομένα.

Όσον αφορά τις μελλοντικές επεκτάσεις της παρούσας εργασίας, θα μπορούσε να εξεταστεί η εφαρμογή πιο σύνθετων αρχιτεκτονικών νευρωνικών δικτύων ή και η ενσωμάτωση attention mechanisms, όπως τα Transformer-based μοντέλα, για τη βελτίωση της απόδοσης σε πιο απαιτητικά tasks. Παράλληλα, η εφαρμογή των έμμεσων διεργασιών σε μεγαλύτερα και πιο ετερογενή σύνολα δεδομένων θα μπορούσε να δώσει σημαντικά αποτελέσματα σχετικά με την γενικευσιμότητα των χαρακτηριστικών που μαθαίνουν τα μοντέλα.

Μια ακόμη σημαντική κατεύθυνση για περαιτέρω μελέτη θα ήταν η εξέταση της προσαρμογής των έμμεσων διεργασιών σε δεδομένα διαφορετικής φύσης, όπως δεδομένα βίντεο ή ακόμα και δεδομένα αισθητήρων, για τη βελτίωση της ανάλυσης πολυδιάστατων και διαδοχικών πληροφοριών. Τέλος, η χρήση πιο προηγμένων τεχνικών fine-tuning, όπως το progressive fine-tuning, θα μπορούσε να οδηγήσει σε μεγαλύτερη αποδοτικότητα και οικονομία πόρων, ειδικά σε περιπτώσεις όπου τα δεδομένα του downstream task είναι περιορισμένα.

Η έρευνα σε αυτό τον τομέα παρουσιάζει σημαντικές προοπτικές, και η παρούσα εργασία προσέφερε μία λεπτομερή αξιολόγηση των δυνατοτήτων της Αυτο-Εποπτευόμενης Μάθησης με τη χρήση έμμεσων διεργασιών στη βελτίωση της απόδοσης νευρωνικών δικτύων για την ταξινόμηση εικόνων.

1

Βιβλιογραφία

Βιβλια:

- Μπούταλης, Ι. & Συρακούλης, Γ. (2010). Υπολογιστική Νοημοσύνη και εφαρμογές. Κρίκος – Αφοί Παπαμάρκου Ο.Ε.
- Goodfellow, I. (2016). Deep learning. MIT Press Ltd
- Hecht-Nielsen, R. (1989). *Neurocomputing*. Addison-Wesley Longman Publishing Co., Inc..

Αναφορές:

[1] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

[2] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

[3] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[5] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[6] Tan, M. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*.

[7] He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9729-9738).

[8] Doersch, C., Gupta, A., & Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision* (pp. 1422-1430).

[9] Noroozi, M., & Favaro, P. (2016, September). Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision* (pp. 69-84). Cham: Springer International Publishing.

[10]Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255). Ieee.

[11]Zhang, R., Isola, P., & Efros, A. A. (2016). Colorful image colorization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*

14 (pp. 649-666). Springer International Publishing.

[12]Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2536-2544).

[13]Gidaris, S., Singh, P., & Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.

[14]Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.

[15]Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345-1359.

[16]Long, M., Cao, Y., Wang, J., & Jordan, M. (2015, June). Learning transferable features with deep adaptation networks. In *International conference on machine learning* (pp. 97-105). PMLR.

[17]Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

[18]McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*, 115-133.

[19]Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[20]Ioffe, S. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

[21]Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929-1958.

[22]Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

[23]Lowe, D., & Broomhead, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex systems*, *2*(3), 321-355.

[24]Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533-536.

[25]Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, *8*, 279-292.

[26]Rummery, G. A., & Niranjan, M. (1994). *On-line Q-learning using connectionist systems* (Vol. 37, p. 14). Cambridge, UK: University of Cambridge, Department of Engineering.

[27]Mnih, K. (2015). Mnih V., Kavukcuoglu K., Silver D., Rusu Aa, Veness J., Bellemare MG, et al. *Human-level control through deep reinforcement learning, Nature*, *518*(7540), 529-533.

[28]Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *58*(1), 267-288.

[29]Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55-67.

[30]Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, *313*(5786), 504-507.

[31]Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2006). Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, *19*.

[32]Masci, J., Meier, U., Cireşan, D., & Schmidhuber, J. (2011). Stacked convolutional auto-encoders for hierarchical feature extraction. In *Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21* (pp. 52-59). Springer Berlin Heidelberg.

[33]Ng, A., & Autoencoder, S. (2011). CS294A Lecture notes. *Dosegljivo: https://web. stanford. edu/class/cs294a/sparseAutoencoder_2011new. pdf.[Dostopano 20. 7. 2016]*.

[34]Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008, July). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning* (pp. 1096-1103).

[35]Kingma, D. P. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[36]Sakurada, M., & Yairi, T. (2014, December). Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis* (pp. 4-11).

[37]Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

[38]Redmon, J. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[39]Ren, S. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*.

[40]Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

[41]Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, *42*, 60-88.

[42]Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., ... & Zieba, K. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

[43]Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*.

[44]Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, *27*.

[45]Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, *32*.

[46]Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). {TensorFlow}: a system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (pp. 265-283).

[47]Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825-2830.

[48]Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020, October). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations* (pp. 38-45).

[49]Howard, J., & Gugger, S. (2020). Fastai: a layered API for deep learning. *Information*, *11*(2), 108.

[50]Devlin, J. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[51]Yenduri, G., Ramalingam, M., Selvi, G. C., Supriya, Y., Srivastava, G., Maddikunta, P. K. R., ... & Gadekallu, T. R. (2024). Gpt (generative pre-trained transformer)–a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *IEEE Access*.

[52]Griffin, G., Holub, A., & Perona, P. (2007). *Caltech-256 object category dataset* (Vol. 10). Pasadena: Technical Report 7694, California Institute of Technology.

[53]Zhang, C., Ranjan, S., & Hansen, J. H. (2018, June). An Analysis of Transfer Learning for Domain Mismatched Text-independent Speaker Verification. In *Odyssey* (pp. 181-186).

[54]Patel, C., Shah, D., & Patel, A. (2013). Automatic number plate recognition system (anpr): A survey. *International Journal of Computer Applications*, *69*(9).

[55]Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P. A., & Bottou, L. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, *11*(12).

[56]Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020, November). A simple framework for contrastive learning of visual representations. In *International conference on machine learning* (pp. 1597-1607). PMLR.

[57]Wu, Z., Xiong, Y., Yu, S. X., & Lin, D. (2018). Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3733-3742).

[58]Blog, A. K. (2015). The unreasonable effectiveness of recurrent neural networks. *URL: http://karpathy. github. io/2015/05/21/rnn-effectiveness/dated May*, *21*, 31.

[59]Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4694-4702).