


Πανεπιστήμιο Πειραιώς
ΕΚΕΦΕ "ΔΗΜΟΚΡΙΤΟΣ"
Δ.Π.Μ.Σ. στην Τεχνητή Νοημοσύνη
Τμήμα Ψηφιακών Συστημάτων Πανεπιστημίου Πειραιώς
Ινστιτούτο Πληροφορικής και Τηλεπικοινωνιών ΕΚΕΦΕ «Δημόκριτος»

Μεταφορά μάθησης σε Βαθιά Νευρωνικά Δίκτυα

Εργασία στο μάθημα "Βαθιά Μηχανική Μάθηση"

Νικόλαος Μακρής - mtn2208

Διδάσκων: Γιαννακόπουλος Θεόδωρος, Μεταδιδακτορικός Ερευνητής ΕΚΕΦΕ Δημόκριτος

Αθήνα, 20-7-2023

Περιεχόμενα

Αρχικά Βήματα

1.1 Περιγραφή προβλήματος

Σκοπός της συγκεκριμένης εργασίας ήταν η χρήση δύο διαφορετικών συνόλων δεδομένων (dataset) και η εκπαίδευση νευρωνικών δικτύων για κάθε ένα από αυτά. Επιπλέον το πρώτο μοντέλο θα έπρεπε να χρησιμοποιηθεί ως βάση για την εφαρμογή της μεθόδου της μεταφοράς μάθησης (transfer learning) κατά την εκπαίδευση του δεύτερου δικτύου. Η επιλογή των συνόλων δεδομένων και του τύπου του προβλήματος (π.χ. παλινδρόμηση, ταξινόμηση κλπ.) που χρησιμοποιήθηκαν ήταν ελεύθερη.

Για τη συγκεκριμένη εργασία, επιλέχθηκε ένα πρόβλημα ταξινόμησης πολλών κλάσεων (multiclass classification), όπου με τη χρήση εικόνων γίνεται διάκριση μεταξύ διαφορετικών ειδών τροφής. Τα δύο datasets που χρησιμοποιήθηκαν είναι το UECFOOD256 [3] και το FOOD101 [1].

Πιο συγκεκριμένα, το πρώτο dataset περιλαμβάνει περίπου 31000 φωτογραφίες 256 διαφορετικών τροφών με διαφορετικό αριθμό φωτογραφιών για κάθε μια από τις 256 κατηγορίες, με τις περισσότερες εξ αυτών να είναι κατηγορίες τροφών που συναντώνται στην Ιαπωνία. Στο δεύτερο σύνολο δεδομένων εντοπίζονται 101 κλάσεις με 1000 ακριβώς φωτογραφίες ανά κλάση.

Τα δύο σύνολα έχουν κάποια κοινά στοιχεία, πχ υπάρχει η κλάση ”πίτσα” και στα δύο, αλλά η μεγάλη πλειοψηφία των κλάσεων ανήκει αποκλειστικά σε ένα από τα δύο σύνολα. Η λογική που ακολουθήθηκε ήταν να επιλεγεί το πρώτο dataset και να εκπαιδευτεί πάνω σε αυτό ένας ταξινομητής (classifier) με όσο το δυνατόν μεγαλύτερη ακρίβεια πρόβλεψης του είδους της τροφής και εν συνεχεία να χρησιμοποιηθεί για μεταφορά μάθησης στο δεύτερο dataset. Στον πίνακα 1.1 παρατίθενται τα στοιχεία για το μέγεθος και τον αριθμό των κλάσεων των συνόλων δεδομένων που χρησιμοποιήθηκαν.

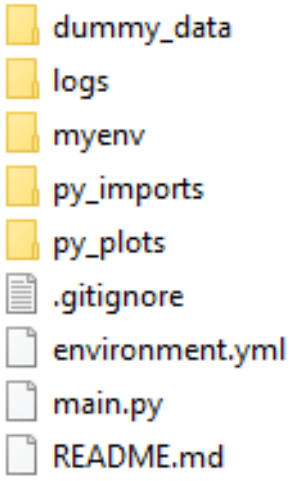
	UECFOOD256	FOOD101
Κλάσεις	256	101
Αριθμός εικόνων	31651	101000

Πίνακας 1.1: Σύνολα δεδομένων (datasets)

1.2 Περιγραφή παραδοτέου προγράμματος

Πριν την αναλυτική περιγραφή των βημάτων που ακολουθήθηκαν για την εκπαίδευση των νευρωνικών δικτύων παρατίθεται μια σύντομη περιγραφή της δομής του παραδοτέου πακέτου κώδικα ώστε να υπάρχει μια εικόνα της σύνδεσης όλων όσων περιγράφονται στο υπόλοιπο κείμενο. Στο σχήμα 1.1 φαίνεται η δομή του παραδοτέου προγράμματος. Τα βασικά στοιχεία συνοψίζονται στα εξής.

1. dummy_data: Φάκελος που περιέχει ένα πολύ μικρό αριθμό δεδομένων, ώστε να φαίνεται η δομή που χρησιμοποιείται κατά την εκτέλεση του προγράμματος
2. log_tensorboard & logs : Φάκελοι οι οποίοι δημιουργούνται κατά τη διάρκεια της πρώτης εκτέλεσης του προγράμματος και περιέχουν log αρχεία με χρήσιμες πληροφορίες για την πορεία εκτέλεσης του προγράμματος
3. myenv : Φάκελος που περιέχει το εικονικό περιβάλλον το οποίο χρησιμοποιήθηκε για την ανάπτυξη του κώδικα. Η ενεργοποίηση του γίνεται με βάση τις οδηγίες που υπάρχουν στο README αρχείο
4. py_imports : Φάκελος που περιέχει τα διάφορα python scripts (κλάσεις, συναρτήσεις) που χρησιμοποιούνται στο κυρίως πρόγραμμα
5. py_plots : Φάκελος που περιέχει τις συναρτήσεις (κώδικας python) που χρησιμοποιούνται για τη δημιουργία διαφόρων σχημάτων και γραφικών παραστάσεων που χρησιμοποιούνται στην παρούσα αναφορά
6. results : Φάκελος αποτελεσμάτων που δημιουργείται κατά την πρώτη εκτέλεση του προγράμματος
7. main.py : Κυρίως πρόγραμμα python
8. .gitignore, README, environment : Αρχεία του git και του εικονικού περιβάλλοντος



Σχήμα 1.1: Δομή προγράμματος

1.3 Προεπεξεργασία δεδομένων

Οι εικόνες και των δύο dataset είναι διαφόρων διαστάσεων και αναλογιών πλάτους-ύψους, οπότε πριν την εκκίνηση της εκπαίδευσης παρεμβάλλεται υποχρεωτικά ένα βήμα προεπεξεργασίας κατά το οποίο όλες οι εικόνες αποκτούν τις ίδιες διαστάσεις. Για τον σκοπό αυτό αναπτύχθηκε κώδικας ο οποίος αναλαμβάνει την μετατροπή των εικόνων στο επιθυμητό τελικό μέγεθος.

Επιπλέον, στο σύνολο δεδομένων UECFOOD256 δίνονται τα περιγράμματα (bounding boxes) που περικλείουν το είδος της τροφής προς ταξινόμηση και αποκλείουν όλο το περιβάλλον. Γίνεται λοιπόν χρήση μιας μεθόδου αποκοπής (cropping), ώστε να μείνει μόνο το συγκεκριμένο αντικείμενο εντός εικόνας και να διευκολυνθεί η διαδικασία της εκπαίδευσης επιτυγχάνοντας τελικά μεγαλύτερη ακρίβεια πρόβλεψης. Στην εικόνα 1.2 φαίνεται το αποτέλεσμα της αποκοπής σε μια εικόνα του UECFOOD256. Σε όλες τις περιπτώσεις επιλέχθηκε η διατήρηση και των τριών καναλιών χρωμάτων σε κάθε εικόνα και η μη μετατροπή τους σε μονοχρωματικές για λόγους επίτευξης μεγαλύτερης ακρίβειας κατά τη διάρκεια της εκπαίδευσης, παρά το αυξημένο υπολογιστικό κόστος που αυτή η απόφαση συνεπάγεται.

(α') Πριν (β') Μετά

Σχήμα 1.2: UECFOOD256 αποκοπή εικόνας

Επιπλέον η ύπαρξη των bounding boxes ήταν και ο βασικός λόγος που επιλέχθηκε το συγκεκριμένο σύνολο δεδομένων ως αρχικό και όχι το FOOD101, όπως ήταν η αρχική πρόθεση, καθώς προτιμήθηκε η λογική της εκπαίδευσης του μοντέλου σε ένα ”καθαρό” σύνολο δεδομένων το οποίο περιέχει περισσότερες κλάσεις αλλά παράλληλα λιγότερες εικόνες ανά κλάση, ώστε να εκτεθεί το μοντέλο σε πολλά διαφορετικά είδη τροφών και να αναπτύξει με αυτό τον τρόπο τη δυνατότητα να εντοπίζει συγκεκριμένα χαρακτηριστικά στοιχεία (features), τα οποία και θα μπορούσαν να χρησιμοποιηθούν σε ένα άλλο σύνολο δεδομένων για την ταξινόμηση τροφών.

Αντίθετα στο δεύτερο σύνολο δεδομένων δεν είναι διαθέσιμα αυτά τα πλαίσια εντοπισμού και για αυτό τον λόγο δεν είναι δυνατή η απομόνωση και αποκοπή της εικόνας του φαγητού, ενώ αναφέρεται ότι έχει γίνει και λανθασμένη απόδοση κλάσης σε μερικές από τις εικόνες, πράγμα που επηρεάζει αρνητικά το τελικό αποτέλεσμα.

Ένα ακόμη στοιχείο που αξίζει σχολιασμού είναι ότι η προ-επεξεργασία των εικόνων είναι μια αρκετά χρονοβόρα διαδικασία η οποία ξεπερνάει τη μια ώρα για κάθε σύνολο δεδομένων, οπότε επελέγη οι εικόνες να επεξεργαστούν μια φορά για κάθε επιλεγόμενη διάσταση πχ (160, 160, 3) και να αποθηκευτούν σε αρχεία τύπου .h5. Στη συνέχεια κάθε φορά που το πρόγραμμα εκτελείτο οι εικόνες διαβάζονταν απευθείας από τα αρχεία και εισάγονταν στο πρόγραμμα ως πίνακες (numpy.arrays) τύπου int8.

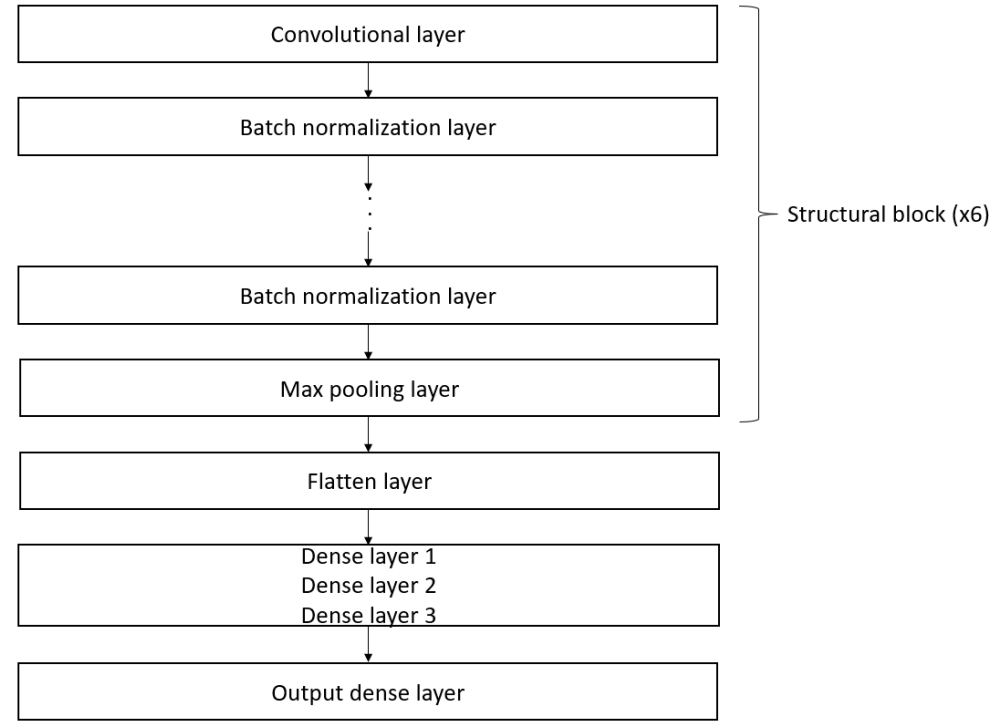
Τέλος μετά την προ-επεξεργασία και αποθήκευση των εικόνων γίνεται ο διαχωρισμός των δεδομένων σε training-validation-test set με αναλογίες 70%-15%-15% και είναι πλέον δυνατή η εκτέλεση του επόμενου βήματος που αφορά την εκπαίδευση του νευρωνικού δικτύου.

Εκπαίδευση μοντέλων

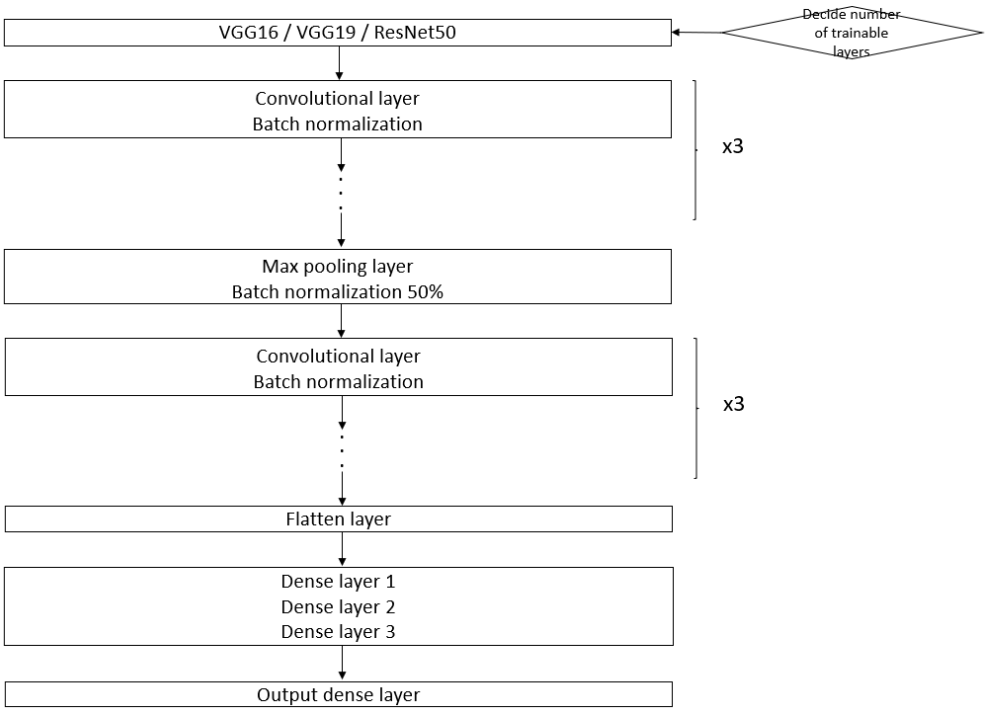
2.1 Αρχιτεκτονικές

Με την ολοκλήρωση της προ-επεξεργασίας των εικόνων είναι εφικτό να προχωρήσει η εκπαίδευση των διαφόρων μοντέλων ταξινομητών για το πρώτο σύνολο δεδομένων. Λόγω της φύσης του προβλήματος και τη χρήση νευρωνικών δικτύων, κάθε εποχή εκπαίδευσης είναι αρκετά χρονοβόρα και οι δοκιμές των διαφόρων αρχιτεκτονικών είναι εκ των πραγμάτων περιορισμένες.

Στην παρούσα εργασία έγινε δοκιμή τεσσάρων διαφορετικών αρχιτεκτονικών, από τις οποίες η πρώτη δημιουργήθηκε από την αρχή, ενώ οι υπόλοιπες δημιουργήθηκαν ως ένα μείγμα προ-εκπαιδευμένων μοντέλων, που βασίζονταν στις γνωστές αρχιτεκτονικές VGG16, VGG19 και ResNet50, καθώς και ένα επιπλέον τμήμα συνελικτικών (CNN) και διασυνδεδεμένων (fully connected) επιπέδων (layers) τα οποία προστέθηκαν εκ των υστέρων ώστε μέσω αυτού να γίνει ο εντοπισμός των πιο λεπτομερών χαρακτηριστικών του προβλήματος. Οι δομές των αρχιτεκτονικών φαίνονται στα σχήματα 2.1 και 2.2 αντίστοιχα.



Σχήμα 2.1: Αρχικτετονική C15D3net



Σχήμα 2.2: Αρχικτετονικές VGG16mod, VGG19mod, ResNet50mod

2.2 Ρύθμιση παραμέτρων

Μετά την ανάπτυξη των διαφόρων αρχιτεκτονικών μπορεί να ακολουθήσει η διεξαγωγή δοκιμών ώστε να καταλήξουμε στο ιδανικότερο μοντέλο για την επίλυση του προβλήματος ταξινόμησης.

Αντικειμενικός στόχος αυτών των δοκιμών ήταν η επίτευξη όσο το δυνατόν μεγαλύτερης ακρίβειας πρόβλεψης (accuracy) κατά τη διάρκεια της εκπαίδευσης, αλλά κρισιμότερα κατά την διαδικασία επαλήθευσης (validation), η οποία αποτελεί ένα καλό δείκτη της δυνατότητας γενίκευσης του αναπτυσσόμενου μοντέλου.

Έχοντας αποφασίσει, σε γενικές γραμμές, τη δομή των διαφόρων αρχιτεκτονικών, όπως παρουσιάστηκε στα σχήματα 2.1 και 2.2, ακολουθεί η παρουσίαση των δοκιμών για την επιλογή της τελικής μορφής του νευρωνικού δικτύου. Οι δοκιμές αυτές μπορούν να αποδοθούν στις παρακάτω κατηγορίες.

1. Μέγεθος των εικόνων
2. Χρήση γεννήτριας εικόνων (image generator)
3. Χρήση regularizer L1,L2 για τη μείωση του βαθμού υπερπροσαρμογής (overfitting) στα δεδομένα
4. Επιμέρους παράμετροι εκπαίδευσης (αριθμός εποχών, είδος βελτιστοποιητή, κτλ.) για γρηγορότερη και καλύτερη σύγκλιση του μοντέλου

5. Αριθμός επιπέδων προς εκπαίδευση (trainable layers) στις αρχιτεκτονικές που χρησιμοποιούν προ-εκπαιδευμένα μοντέλα

Μέγεθος εικόνων / Αρχιτεκτονική	(64, 64, 3)	(128, 128, 3)	(160, 160, 3)	(256, 256, 3)
ResNet50mod (1)	√	√	√	√
ResNet50mod (2)			√	
ResNet50mod (3)			√	
ResNet50mod (4)			√	
VGG16mod			√	
VGG19mod			√	
C15D3net			√	

Πίνακας 2.1: Περιπτώσεις αρχιτεκτονικών που ελέγχθηκαν

Στον πίνακα 2.1 παρουσιάζονται οι περιπτώσεις για τις οποίες πραγματοποιήθηκε εκτέλεση του κώδικα. Για τις αρχιτεκτονικές που παρουσιάζονται στην πρώτη στήλη του πίνακα και οι οποίες θα αναφερθούν λεπτομερέστερα παρακάτω ισχύουν τα εξής.

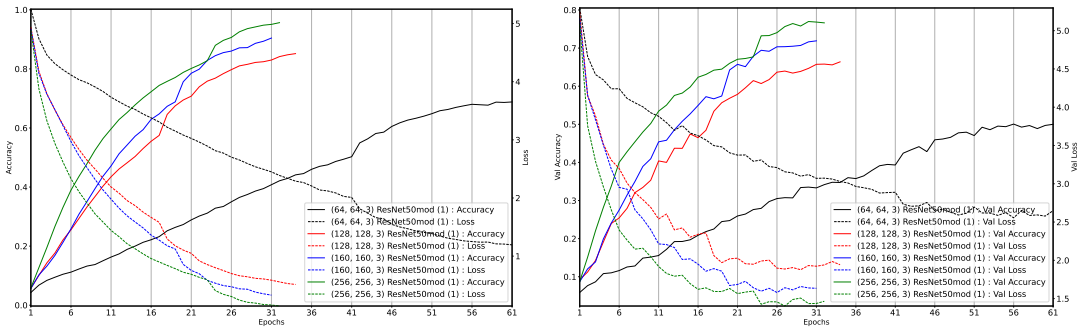
- ResNet50mod (1) : 20 προ-εκπαιδευμένα layers της ResNet50 προς εκπαίδευση, χωρίς εφαρμογή L1/L2 regularization, με χρήση γεννήτριας εικόνων
- ResNet50mod (2) : 20 προ-εκπαιδευμένα layers της ResNet50 προς εκπαίδευση, χωρίς εφαρμογή L1/L2 regularization, χωρίς χρήση γεννήτριας εικόνων
- ResNet50mod (3) : 20 προ-εκπαιδευμένα layers της ResNet50 προς εκπαίδευση, με χρήση L1/L2 regularization, με χρήση γεννήτριας εικόνων
- ResNet50mod (4) : 0 προ-εκπαιδευμένα layers της ResNet50 προς εκπαίδευση, χωρίς εφαρμογή L1/L2 regularization, με χρήση γεννήτριας εικόνων
- VGG16mod : 0 προ-εκπαιδευμένα layers της VGG16 προς εκπαίδευση, χωρίς εφαρμογή L1/L2 regularization, με χρήση γεννήτριας εικόνων
- VGG19mod : 0 προ-εκπαιδευμένα layers της VGG19 προς εκπαίδευση, χωρίς εφαρμογή L1/L2 regularization, με χρήση γεννήτριας εικόνων
- C15D3net : Αρχιτεκτονική που αναπτύχθηκε εξ'αρχής. Αποτελείται από 15 συνελικτικά και 3 πλήρως διασυνδεδεμένα επίπεδα

2.2.0.1 Μέγεθος εικόνων

Αρχικά, η πρώτη παράμετρος που πρέπει να καθοριστεί είναι το μέγεθος των εικόνων που θα χρησιμοποιηθούν κατά την εκπαίδευση. Πρόκειται για μια πολύ σημαντική παράμετρο της οποίας η τιμή επιλέγεται πριν αρχίσει η διαδικασία εκπαίδευσης, κατά το στάδιο της προ-επεξεργασίας εικόνων, αλλά έχει σημαντικές συνέπειες για την πολυπλοκότητα, την τελική ακρίβεια του μοντέλου αλλά και τη δυνατότητα εκπαίδευσης του μοντέλου λόγω επάρκειας μνήμης όπως και τον χρόνο εκπαίδευσης.

Στο σχήμα 2.3α' φαίνεται η ακρίβεια που επιτυγχάνεται κατά τη διάρκεια εκπαίδευσης για την ίδια ακριβώς αρχιτεκτονική, η οποία είναι η τροποποιημένη ResNet50 με 20 trainable layers και χωρίς τη χρήση regularizer, δηλαδή η περίπτωση ResNet50mod (1) που αναφέραμε πιο πάνω, για τέσσερις διαφορετικές διαστάσεις εικόνων (64, 64, 3) (128, 128, 3), (160, 160, 3) και (256, 256, 3).

Στο σχήμα 2.3β', φαίνονται οι τιμές κατά την διαδικασία της επαλήθευσης. Επιπλέον στον πίνακα 2.2 παρουσιάζεται συγκεντρωτικά ο αριθμός των παραμέτρων προς εκπαίδευση για κάθε μια από αυτές τις περιπτώσεις, αλλά και για όλες τις υπόλοιπες περιπτώσεις τις οποίες μελετήσαμε.



(α') Training (β') Validation

Σχήμα 2.3: Επίδραση μεγέθους εικόνας

Αρχιτεκτονική \ Μέγεθος εικόνων	(64, 64, 3)	(128, 128, 3)	(160, 160, 3)	(256, 256, 3)
ResNet50mod (1)	30,221,728	30,477,888	30,561,424	32,403,328
ResNet50mod (2)			30,561,424	
ResNet50mod (3)			30,561,424	
ResNet50mod (4)			21,630,096	
VGG16mod			7,423,621	
VGG19mod			7,423,621	
C15D3net			6,114,716	

Πίνακας 2.2: Αριθμός παραμέτρων προς εκπαίδευση

Όπως είναι φανερό από τον πίνακα 2.2 η χρήση μεγαλύτερων σε μέγεθος εικόνων δεν αυξάνει ιδιαίτερα τον αριθμό των προς εκπαίδευση παραμέτρων, αλλά το σχήμα 2.3 καταδεικνύει ότι η ακρίβεια που επιτυγχάνει το μοντέλο τόσο κατά την εκπαίδευση όσο και κατά την επαλήθευση βελτιώνεται με την αύξηση του μεγέθους, ενώ αντίθετα η συνάρτηση κόστους βαίνει διαρκώς μειούμενη. Αυτό μπορεί να εξηγηθεί από το γεγονός ότι οι μεγαλύτερες σε μέγεθος εικόνες περιέχουν περισσότερες λεπτομέρειες οι οποίες μπορούν να εντοπιστούν κατά την εκπαίδευση του νευρωνικού δικτύου.

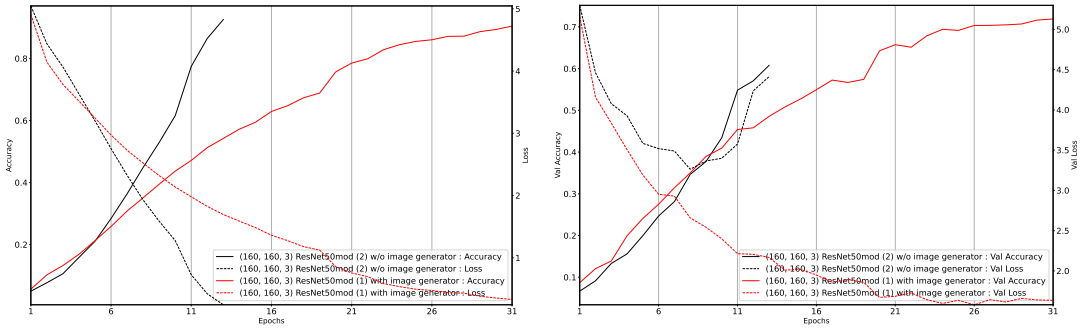
Από την παραπάνω ανάλυση θα περίμενε κανείς να επιλεγούν για την συνέχεια των δοκιμών η χρήση εικόνων (256, 256, 3). Όμως για την σύγκριση μεταξύ μοντέλων αρκεί να έχουν εκπαιδευτεί όλα με τον ίδιο μέγεθος εικόνων οπότε και προτιμήθηκε οι υπόλοιπες δοκιμές να γίνουν με εικόνες διαστάσεων (160, 160, 3).

2.2.0.2 Γεννήτρια εικόνων

Ένα άλλο πολύ σημαντικό στοιχείο για την αποτελεσματική εκπαίδευση του νευρωνικού δικτύου είναι η χρήση γεννήτριας εικόνων (image generator). Πρόκειται επί της ουσίας για μια συνάρτηση βιβλιοθήκης της python η οποία πραγματοποιεί αλλαγές στις εικόνες που χρησιμοποιούνται κατά την εκπαίδευση αλλά και την επαλήθευση. Λόγω της διαφοροποίησης των εικόνων εισόδου, αναμένεται μείωση του βαθμού υπερπροσαρμογής αλλά και αυξημένη ακρίβεια πρόβλεψης, το οποίο επιβεβαιώνεται στο σχήμα 2.4.

Πιο αναλυτικά στο σχήμα 2.4α' το οποίο αναπαριστά την εξέλιξη των τιμών της ακρίβειας και της συνάρτησης κόστους κατά την εκπαίδευση, βλέπουμε ότι και στις δύο περιπτώσεις, με και χωρίς τη χρήση γεννήτριας αριθμών, τα αποτελέσματα είναι παραπλήσια. Το ενδιαφέρον στοιχείο όμως εμφανίζεται από την παρατήρηση του σχήματος 2.4β', όπου όπως και προηγουμένως παρουσιάζεται η ιστορία εξέλιξης της ακρίβειας και της συνάρτησης κόστους για το σύνολο δεδομένων επαλήθευσης. Γίνεται εμφανές παρατηρώντας τις μαύρες καμπύλες, ότι σχετικά γρήγορα κατά τη διάρκεια της εκπαίδευσης χωρίς τη χρήση γεννήτριας εικόνων, το μοντέλο οδηγείται σε υπερπροσαρμογή, καθώς η συνάρτηση κόστους αυξάνει την τιμή και αντίστοιχα η επιτυγχανόμενη ακρίβεια είναι πολύ μικρότερη της ακρίβειας του μοντέλου με χρήση image generator.

Τέλος, στον πίνακα 2.3 φαίνονται τα όρια των επιτρεπόμενων αλλαγών στις εικόνες. Αυτά επιλέχθηκαν με χρήση πολύ λίγων επαναλήψεων και πιθανώς μια συστηματικότερη εξέταση των δυνατών τιμών τους να οδηγούσε σε μικρή βελτίωση των αποτελεσμάτων.



(α') Training (β') Validation

Σχήμα 2.4: Επίδραση χρήσης γεννήτριας εικόνων

Παράμετρος	Τιμή
rotation_range	30
width_shift_range	0.1
height_shift_range	0.1
brightness_range	[0.5, 1.0]
shear_range	0.2
zoom_range	0.2
horizontal_flip	True

Πίνακας 2.3: Παράμετροι image generator

2.2.0.3 Κανονικοποίηση (Regularization)

Μια άλλη σημαντική δοκιμή αφορά την χρήση της κανονικοποίησης των δεδομένων, μέσω της οποίας δύναται να μειωθεί η επίδραση της υπερπροσαρμογής στα νευρωνικά δίκτυα. Λόγω του μεγάλου αριθμού των παραμέτρων προς εκπαίδευση τα νευρωνικά δίκτυα έχουν την τάση να οδηγούνται σε υπερπροσαρμογή. Ο τρόπος με τον οποίο επιτυγχάνεται αυτό στην παρούσα εργασία είναι μέσω της χρήσης επιπέδων απόρριψης (dropout layer) και την κανονικοποίηση L1 και L2.

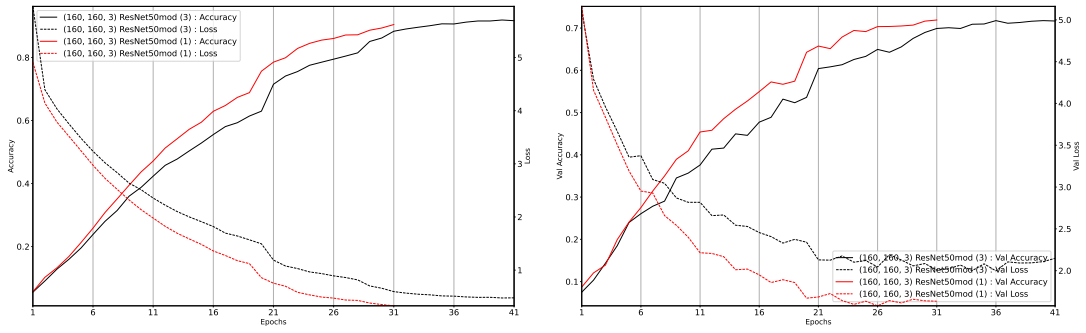
Σχετικά με τα dropout layers αυτά λειτουργούν με τον ακόλουθο τρόπο. Κατά τη διάρκεια της εκπαίδευσης και πιο συγκεκριμένα κατά το βήμα της προώθησης προς τα εμπρός (forward propagation) ορισμένοι από τους νευρώνες του δικτύου απενεργοποιούνται. Έτσι το δίκτυο πρέπει να εξαρτηθεί από άλλους ενεργούς νευρώνες για την αποτελεσματικότερη μάθηση. Πρόκειται δηλαδή για μια μεθοδολογία η οποία χρησιμοποιείται για να μειώσει την ευαισθησία του μοντέλου σε συγκεκριμένα χαρακτηριστικά και να βελτιώσει τη γενίκευση του μοντέλου.

Στην παρούσα αναφορά δεν φαίνονται οι δοκιμές που έγιναν για τον αριθμό των dropout layers αλλά και το ποσοστό των νευρώνων που απορρίπτονται. Παρόλα αυτά σε όλες τις αρχιτεκτονικές του σχήματος 2.2 χρησιμοποιήθηκε ένα dropout layer με 50% ποσοστό απόρριψης.

Οι μέθοδοι κανονικοποίησης L1 και L2 εισάγουν έναν όρο κανονικοποίησης στη συνάρτηση κόστους του μοντέλου, το οποίο οδηγεί σε μηδενισμό ή μείωση της τιμής κάποιων εκ των παραμέτρων του μοντέλου με αποτέλεσμα να μειώνεται η υπερπροσαρμογή του μοντέλου.

Στο σχήμα 2.5 παρουσιάζεται η επίδραση που έχει στην ακρίβεια του μοντέλου η χρήση της κανονικοποίησης. Όπως φαίνεται η εφαρμογή που επιχειρήθηκε δεν είχε κάποιο θετικό αποτέλεσμα, καθώς φαίνεται να οδηγεί σε μικρή σχετικά υποπροσαρμογή το μοντέλο και επιπλέον να οδηγεί το μοντέλο σε περισσότερες επαναλήψεις έως ότου συγκλίνει.

Για αυτό τον λόγο δεν επιλέχθηκε η χρήση L1, L2 regularizer στις υπόλοιπες δοκιμές και στο τελικό μοντέλο.



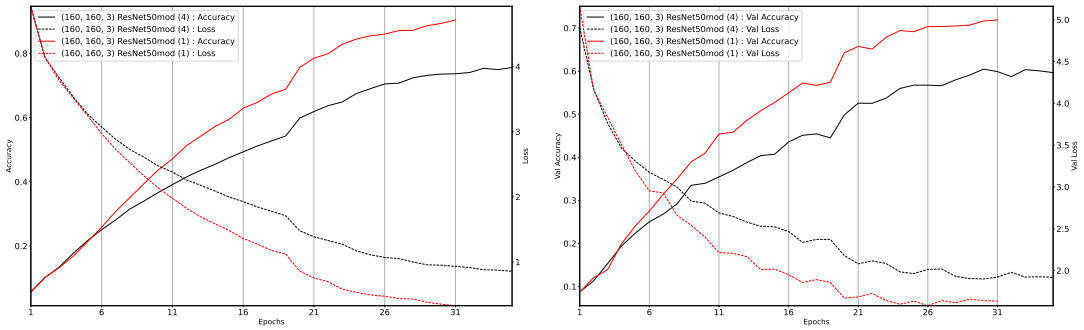
(α') Training (β') Validation

Σχήμα 2.5: Επίδραση χρήσης regularizer

2.2.0.4 Επιμέρους παράμετροι & αριθμός επιπέδων προς εκπαίδευση

Η χρήση διαφορετικών βελτιστοποιητών βρέθηκε πως δεν έχει σημαντική επίδραση στο τελικό αποτέλεσμα, αντίθετα με τον ρυθμό μάθησης (learning rate) και πιο συγκεκριμένα τον τρόπο με τον οποίο αυτός μειώνεται όταν το validation loss σταματάει να βελτιώνεται με το πέρασμα των εποχών εκπαίδευσης.

Για να αυξήσουμε όσο το δυνατόν περισσότερο την ακρίβεια πρόβλεψης δοκιμάστηκε το ξεπάγωμα κάποιων επιπέδων στο προ-εκπαιδευμένο δίκτυο. Πρόκειται δηλαδή για την περίπτωση ResNet50mod (4), όπως αυτή αναφέρεται στους πίνακες 2.1 και 2.2. Στην περίπτωση του ξεπαγώματος επιπέδων όπως φαίνεται και στο σχήμα 2.6 η αύξηση των επιπέδων προς εκπαίδευση οδηγεί σε βελτίωση της ακρίβειας το οποίο είναι αναμενόμενο από τη στιγμή που υπάρχουν περισσότερες προς εκπαίδευση παράμετροι. Πρόκειται για τη λογική που χρησιμοποιείται στο transfer learning.



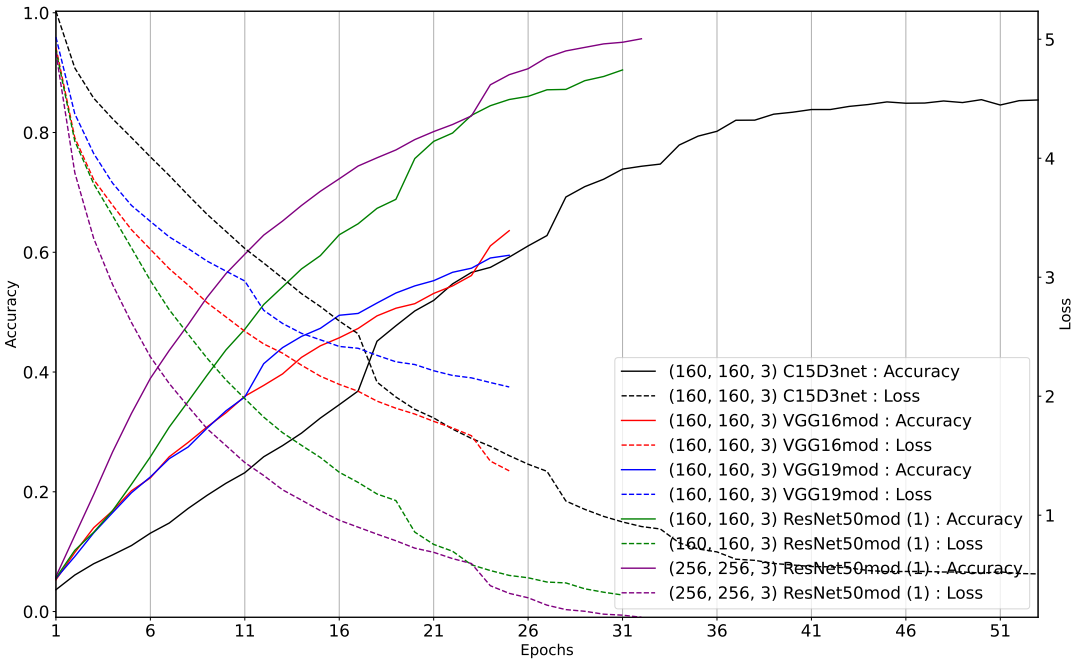
(α') Training (β') Validation

Σχήμα 2.6: Επίδραση ξεπαγώματος επιπέδων

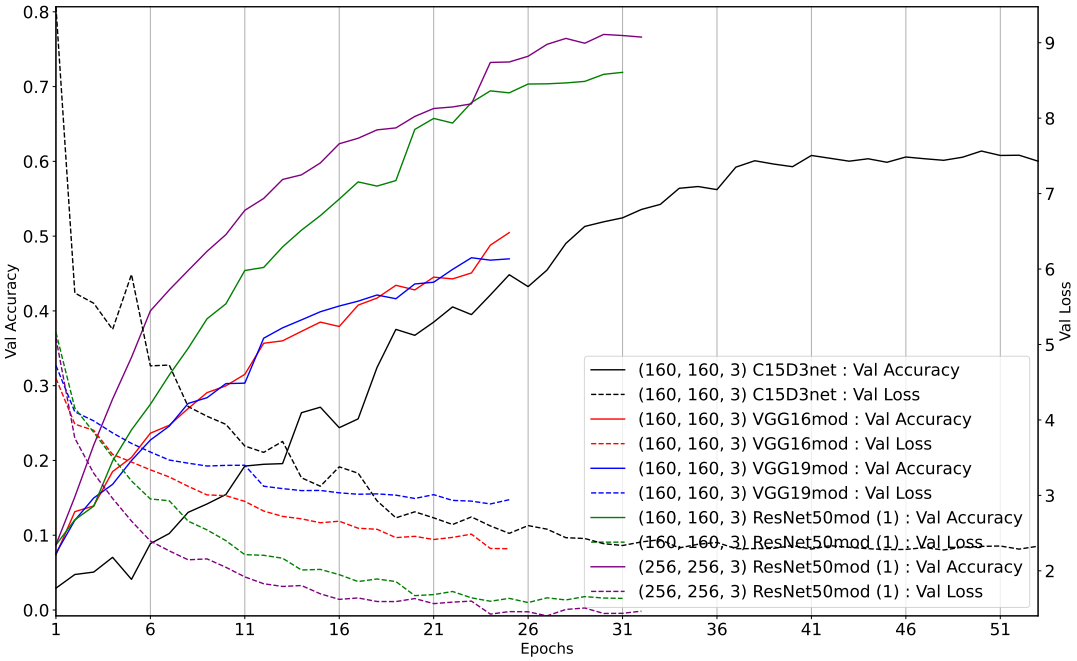
2.2.1 Επιλογή αρχιτεκτονικής

Στις προηγούμενες παραγράφους παρουσιάστηκε η επίδραση που είχε στην εκπαίδευση μιας συγκεκριμένης αρχιτεκτονικής ένα πλήθος παραμέτρων όπως το μέγεθος των εικόνων, η χρήση image generator, καθώς και οι αποφάσεις που λήφθηκαν βασιζόμενοι σε αυτές τις δοκιμές. Η υπόθεση που έγινε ήταν ότι η συμπεριφορά που επιδείχθηκε στο τροποποιημένο μοντέλο ResNet50 θα ήταν ίδια και στα υπόλοιπα μοντέλα.

Έχοντας καταλήξει λοιπόν στο ποιο είναι το καταλληλότερο ResNet50mod, παρουσιάζουμε στο σχήμα 2.7 την εξέλιξη των τιμών ακρίβειας και της συνάρτησης κόστους για πέντε μοντέλα. Όπως γίνεται αντιληπτό το μοντέλο με την μεγαλύτερη ακρίβεια είναι το ResNet50mod για εικόνες (256, 256, 3) το οποίο και θα χρησιμοποιήσουμε στο επόμενο βήμα της εργασίας της μεταφοράς μάθησης.



(α') Training



(β') Validation

Σχήμα 2.7: Ιστορία Εκπαίδευσης

Μεταφορά μάθησης

3.1 Αρχιτεκτονική

Όπως έχει ήδη αναφερθεί η εκπαίδευση του μοντέλου για την πρόβλεψη των εικόνων τροφής του συνόλου δεδομένων FOOD101 έγινε με τη χρήση της τεχνικής της μεταφοράς μάθησης. Για τον λόγο αυτό χρησιμοποιήθηκε το εκπαιδευμένο μοντέλο ResNet50mod για εικόνες εισόδου μεγέθους (256, 256, 3) και πάνω σε αυτό έγιναν οι αλλαγές που απαιτούνταν ώστε να προχωρήσει η εκπαίδευσή του.

Η μορφή της αρχιτεκτονικής αυτού του μοντέλου φαίνεται στο σχήμα 3.1, το οποίο μοιάζει πολύ με αυτό του σχήματος 2.2 με ουσιαστικές όμως διαφορές που οπτικοποιούνται από τα διαφορετικά χρώματα των πλαισίων του σχήματος.

Πιο αναλυτικά, με μαύρο χρώμα παριστάνονται τα τμήματα της αρχιτεκτονικής του δικτύου που μένουν ακριβώς ίδια όπως αυτά εκπαιδεύτηκαν κατά την ανάπτυξη του μοντέλου ResNet50mod. Με κόκκινο χρώμα φαίνονται τα τμήματα τα οποία αλλάζουν προαιρετικά και για τα οποία όπως φαίνεται στο σχήμα 3.2 εκτελέστηκαν δοκιμές. Πρόκειται για το επιπλέον τμήμα συνελικτικών επιπέδων που είχε προστεθεί επί του προ-εκπαιδευμένου ResNet50. Τέλος, με μπλε χρώμα φαίνονται τα επίπεδα τα οποία είναι καινούρια και τα οποία πρακτικά αντικαθιστούν τα εξωτερικά πλήρως διασυνδεδεμένα επίπεδα της ResNet50mod.

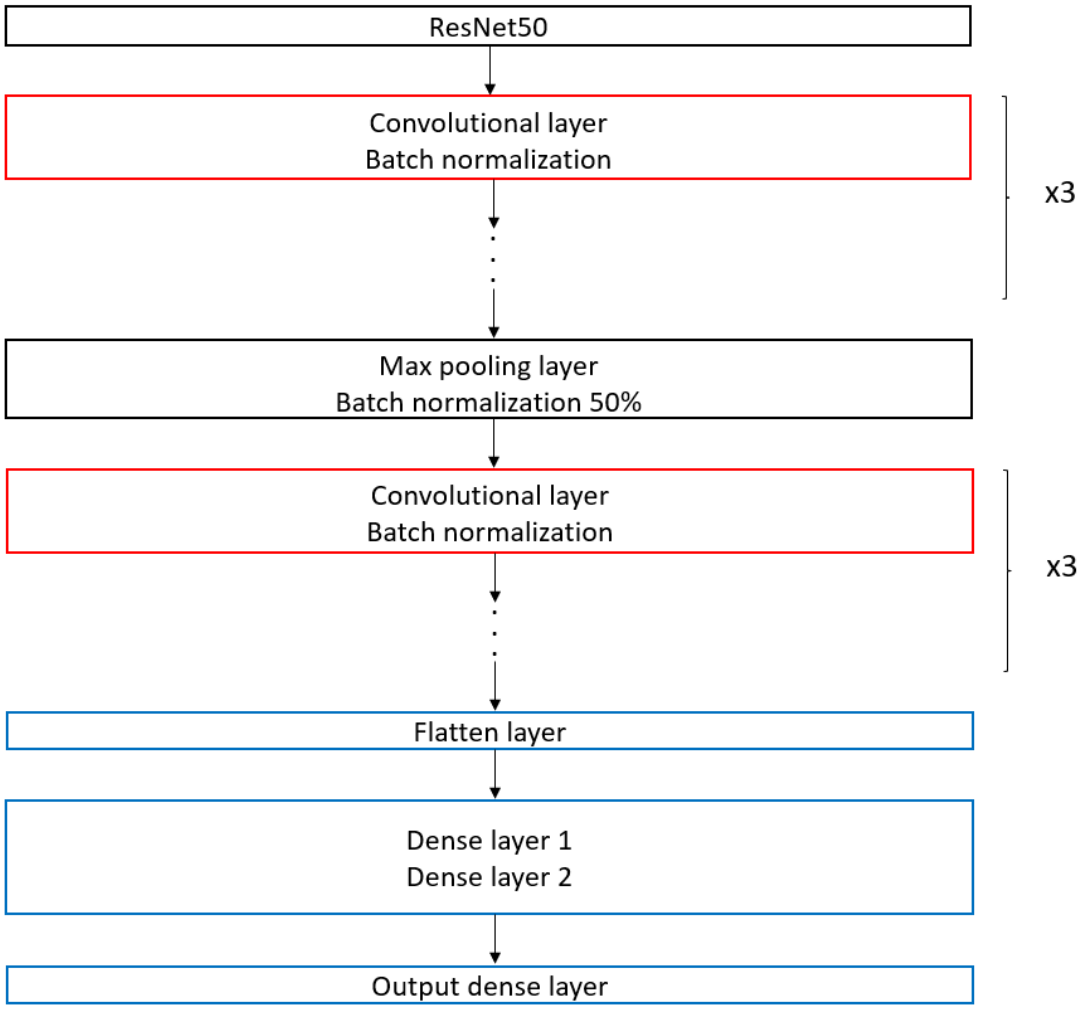
Με βάση λοιπόν αυτήν την αρχιτεκτονική εκτελέστηκαν τρεις διαφορετικές αξιολογήσεις οι οποίες συνοψίζονται παρακάτω, ενώ σχηματοποιούνται στο σχήμα 3.2.

1. ResNet50mod_transfer (1) : 0 ξεπαγωμένα layers σε σχέση με την αρχιτεκτονική ResNet50mod
2. ResNet50mod_transfer (2) : Ξεπάγωμα των τριών εξωτερικών συνελικτικών layers σε σχέση με την αρχιτεκτονική ResNet50mod
3. ResNet50mod_transfer (3) : Ξεπάγωμα και των έξι εξωτερικών συνελικτικών layers σε σχέση με την αρχιτεκτονική ResNet50mod

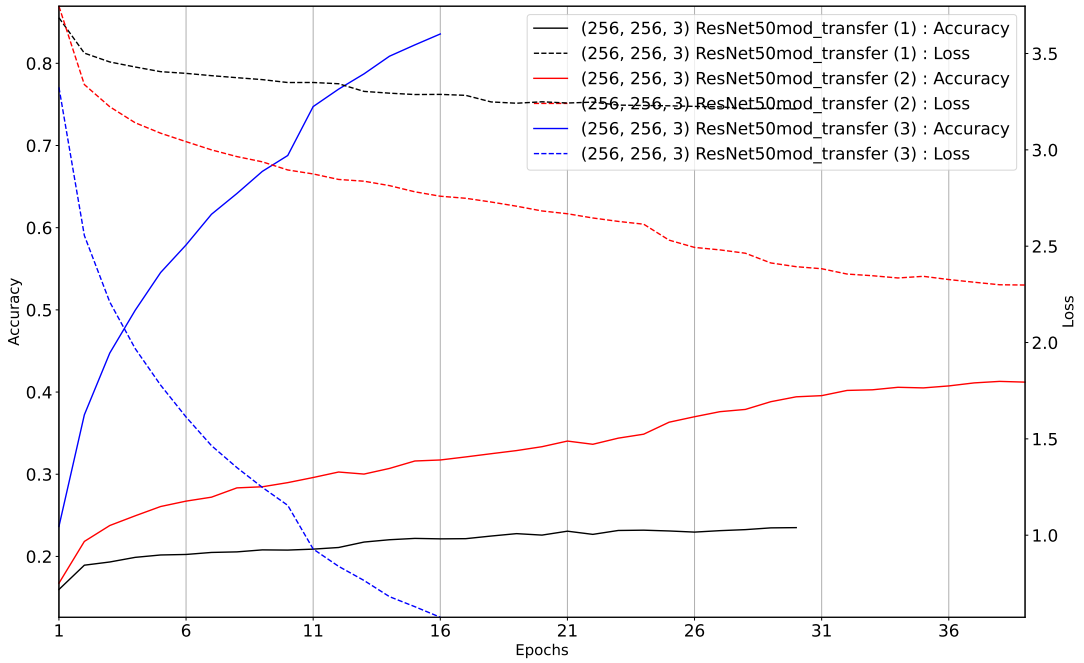
Τέλος δίνεται και ο πίνακας 3.1, όπου φαίνεται ο αριθμός των προς εκπαίδευση παραμέτρων για κάθε μια από τις τρεις περιπτώσεις, η οποία εξηγεί και τον λόγο που το ξεπάγωμα των έξι συνελικτικών επιπέδων οδηγεί σε μεγαλύτερη ακρίβεια πρόβλεψης.

Περίπτωση	Αριθμός παραμέτρων προς εκπαίδευση
ResNet50mod_transfer (1)	570,469
ResNet50mod_transfer (2)	9,225,829
ResNet50mod_transfer (3)	23,386,213

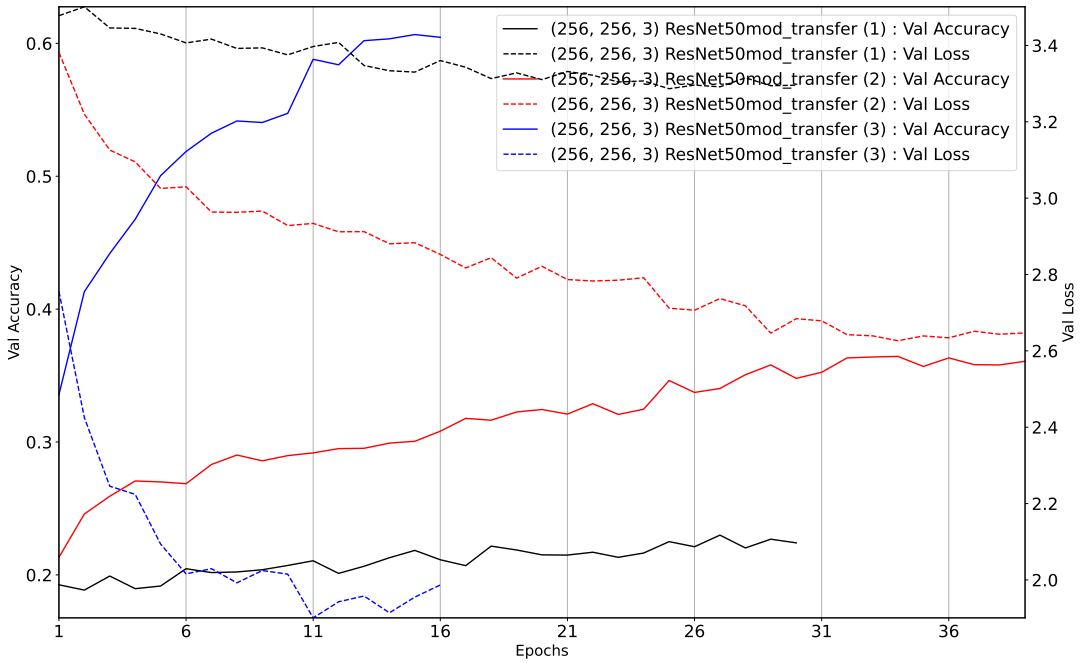
Πίνακας 3.1: Αριθμός προς εκπαίδευση παραμέτρων



Σχήμα 3.1: Αρχτιτεκτονική ResNet50mod (transfer learning)



(α') Training



(β') Validation

Σχήμα 3.2: Ιστορία εκπαίδευσης transfer learning

Αποτελέσματα

4.1 Αποτελέσματα

Στα τρία προηγούμενα κεφάλαια παρουσιάστηκε η διαδικασία εκπαίδευσης δύο νευρωνικών δικτύων ικανών να διακρίνουν και να ταξινομήσουν διαφορετικά είδη τροφών με τη χρήση εικόνων. Όπως αναφέρθηκε τα μοντέλα που επιλέξαμε ήταν το ResNet50mod (1) και το ResNet50mod_transfer (3).

Στον παρακάτω πίνακα 4.1 φαίνονται τα αποτελέσματα για την ακρίβεια, λεπτομέρεια (precision), ευαισθησία (recall) και την ακρίβεια top-5 για τα δύο μοντέλα. Επιπλέον στον ίδιο πίνακα αναγράφεται και η ακρίβεια αναφοράς (baseline accuracy) , η οποία ορίζεται ως η ακρίβεια που θα επιτύγχανε ένας αλγόριθμος που θα απέδιδε σε όλες τις παρατηρήσεις την πιο συχνά εμφανιζόμενη κατά την εκπαίδευση κλάση. Σαν δεδομένα ελέγχου χρησιμοποιήθηκαν αυτά του test συνόλου δεδομένων που δεν είχαν χρησιμοποιηθεί μέχρι αυτό το τελικό στάδιο.

Πρέπει να σημειωθεί ότι ενώ η ακρίβεια μπορεί να υπολογιστεί απευθείας διαιρώντας τις σωστές προβλέψεις με το σύνολο των προς πρόβλεψη σημείων, ο υπολογισμός για τους άλλους δύο δείκτες δεν είναι τόσο απλός. Αυτό γιατί πρέπει να υπολογιστεί η λεπτομέρεια (ή ισοδύναμα η ευαισθησία) για κάθε κλάση και στη συνέχεια παίρνοντας τον σταθμισμένο μέσο όρο να υπολογιστεί για το σύνολο του προβλήματος. Οι σχέσεις υπολογισμού αυτών των τριών δεικτών δίνονται από τις σχέσεις 4.1.

Αναφορικά με τον ορισμό της ακρίβειας top-5, χρησιμοποιείται το ακόλουθο. Ο αλγόριθμος πρόβλεψης επιστρέφει τις πιθανότητες η κάθε εικόνα του test να ανήκει σε κάποια από τις 256 για το πρώτο ή 101 για το δεύτερο πρόβλημα κλάσεις. Έτσι αν η σωστή απάντηση βρίσκεται σε κάποια από τις πέντε επικρατέστερες κλάσεις τότε η απάντηση θεωρείται σωστή.

Κλείνοντας στο σχήμα 4.1 δίνεται ο πίνακας σύγχυσης (confusion matrix) για τα test dataset του πρώτου προβλήματος και αντίστοιχα στο σχήμα 4.2 του δεύτερου προβλήματος.

$$ACC = \frac{TP + TN}{P + N} \tag{4.1.1a}$$

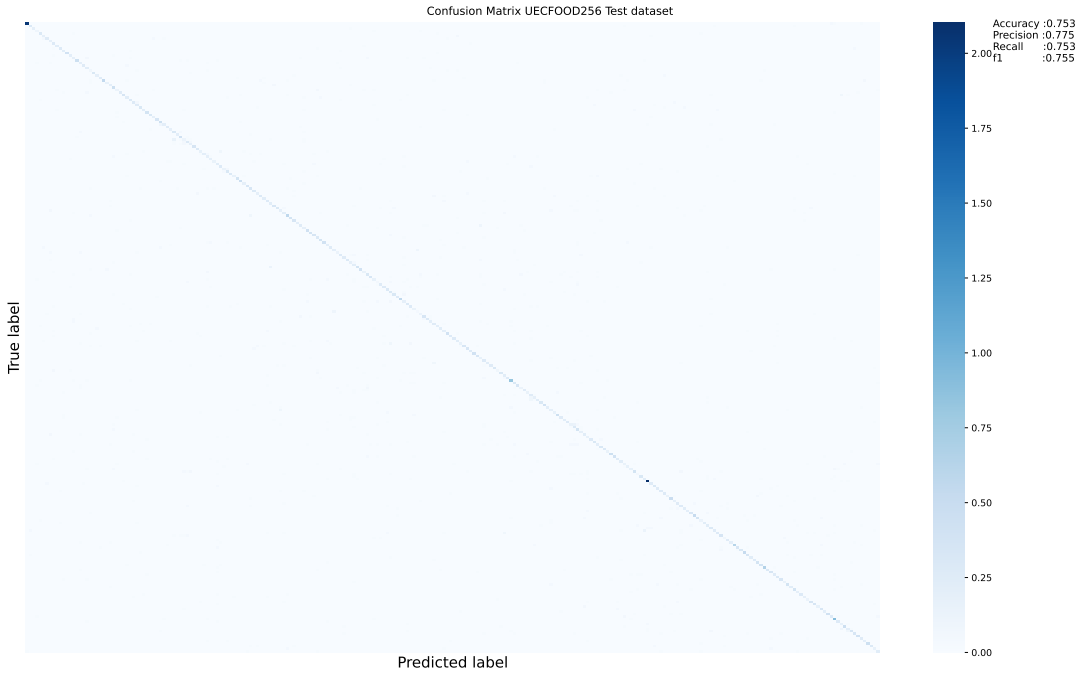
$$PPV_{class} = \frac{TP}{TP + FP} \tag{4.1.1b}$$

$$SEN_{class} = \frac{TP}{TP + FN} \tag{4.1.1c}$$

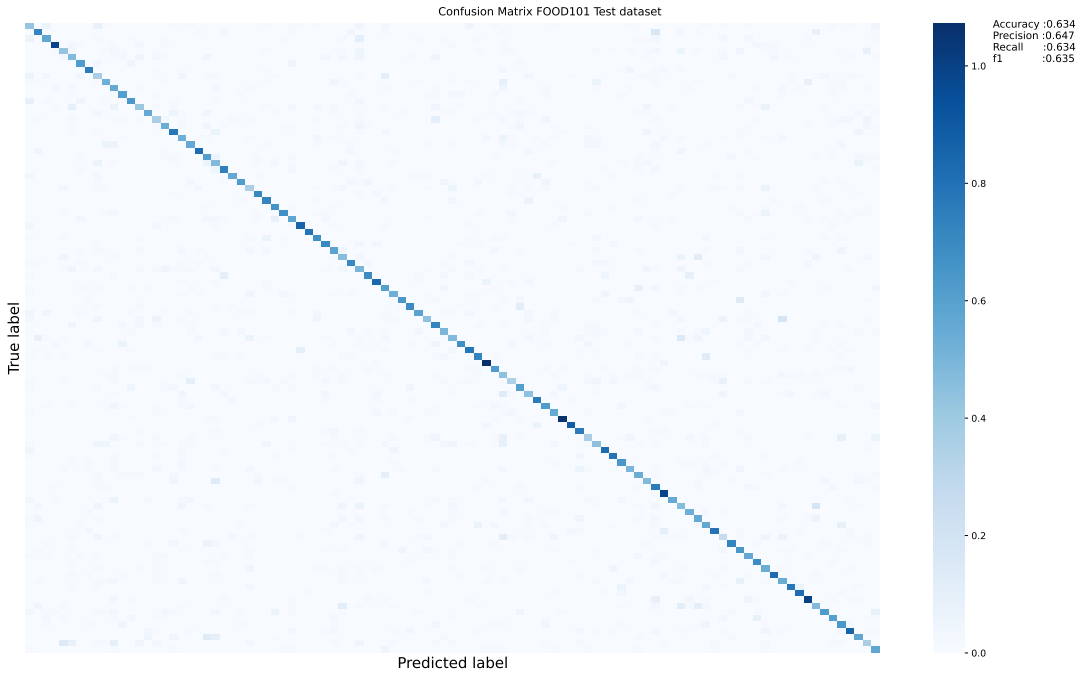
$$\begin{cases} TP : True\,Positive,\ FP : False\,Positive \\ TN : True\,Negative,\ FN : False\,Negative \end{cases} \tag{4.1.1e}$$

UECFOOD256		FOOD101	
Baseline Accuracy	0.02186	Baseline Accuracy	0.01386
Accuracy	0.76751	Accuracy	0.63415
Top-5 Accuracy	0.90764	Top-5 Accuracy	0.82607
Precision	0.78941	Precision	0.64691
Recall	0.76751	Recall	0.63534

Πίνακας 4.1: Τελικά αποτελέσματα



Σχήμα 4.1: Confusion matrix ResNet50mod



Σχήμα 4.2: Confusion matrix ResNet50mod_transfer

4.2 Συμπεράσματα

Οι προκλήσεις που αντιμετωπίστηκαν κατά τη διάρκεια της εκπαίδευσης οφείλονταν σε μια σειρά παραμέτρων και περιγράφηκαν σε διάφορα σημεία της αναφοράς. Συνοπτικά μπορούν να κατηγοριοποιηθούν στις παρακάτω κατηγορίες.

1. Επάρκεια υπολογιστικών πόρων (μνήμης RAM) και χρόνου εκτέλεσης
2. Πλήθος υπερπαραμέτρων και εντοπισμός βέλτιστης λύσης
3. Προλήματα overfitting και underfitting

Αναλυτικότερα, η επάρκεια υπολογιστικών πόρων είναι σημαντική για όλα τα προβλήματα deep learning καθώς ο όγκος των δεδομένων είναι ιδιαίτερα μεγάλος με αποτέλεσμα ένας απλός οικιακός υπολογιστής να μην μπορεί να ανταπεξέλθει. Το πρόβλημα δεν αφορά μόνο την ταχύτητα επεξεργασίας των δεδομένων, αλλά επίσης και τις απαιτήσεις σε μνήμη RAM. Για τον λόγο αυτό, στην παρούσα εργασία ήταν αδύνατον να χρησιμοποιηθεί κάποιο υπάρχον διαδικτυακό μέσον που εκτελεί τον κώδικα παράλληλα, καθώς η RAM ξεπερνούσε τα 25-30 GB με αποτέλεσμα να μην μπορεί να εκτελεστεί. Έτσι καταφύγαμε στη λύση του οικιακού υπολογιστή με αποτέλεσμα να μην παραλληλοποιηθεί η διαδικασία μας.

Το πλήθος των υπερπαραμέτρων προς προσδιορισμό (tuning) είναι πολύ μεγάλο και όπως είδαμε λόγω και του αυξημένου χρόνου είναι αδύνατο να κάνουμε ακριβή προσδιορισμό

του βέλτιστου συνδυασμού. Κατά συνέπεια, μελετήθηκαν κάποιες ενδεικτικές περιπτώσεις, οπότε είναι πιθανό ότι μια πιο ενδελεχής δοκιμή υπερπαραμέτρων θα οδηγούσε σε ακόμη καλύτερα αποτελέσματα.

Τα προβλήματα overfitting και underfitting ήταν παρόντα σε διάφορες περιπτώσεις. Έτσι όταν δεν είχαμε πολλά layers προς εκπαίδευση δεν πετυχαίναμε επαρκή ακρίβεια, ενώ αντίθετα όσο αυξάναμε τα layers αύξανε η ακρίβεια αλλά υπήρχε overfitting. Για αυτό το λόγο, χρησιμοποιήσαμε dropout layers το οποίο φάνηκε να βοηθάει τα τελικά αποτελέσματα. Εκ νέου όμως το υπολογιστικό κόστος ήταν απαγορευτικό για μια βαθύτερη μελέτη.

Βιβλιογραφία

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

[2] Aurélien Géron. *Hands-On Machine learning with Scikit-Learn, Keras & TensorFlow*. O'REILLY, 2nd edition, September 2019.

[3] https://www.kaggle.com/datasets/rkuo2000/uecfood256.