Τμήμα Πληροφορικής και Τηλεματικής
Χαροκόπειο Πανεπιστήμιο
ΕΠ34 Εφαρμογές Τεχνητής Νοημοσύνης

2η Άσκηση: Συνελικτικά Νευρωνικά Δίκτυα

ΕΙΡΗΝΗ ΜΑΡΙΑ ΛΥΚΟΥΔΗ ΑΜ:2021050

2.1 Κλάση COVID19Dataset

Αρχικοποίηση της κλάσης COVID19Dataset : def __init__(self, rootdir, transform=None):

- **Είσοδοι**:
  - rootdir: το pathπου περιέχει τις εικόνες οργανωμένες σε υποφακέλους ανά κάθε κατηγορία
  - transform: Μετασχηματισμοί που εφαρμόζονται στις εικόνες (π.χ.αν υπάρξει κάποια αλλαγή μεγέθους, μετατροπή σε Tensor).
- **Λειτουργίες**:
  - Δημιουργεί μια λίστα self.data που περιέχει ζεύγη (path εικόνας// αριθμό κατηγορίας)
  - Οι κατηγορίες ορίζονται στην self.classes ως: ['COVID', 'Lung_Opacity', 'Normal', 'Viral Pneumonia'].
  - Για κάθε φάκελο κατηγορίας, ελέγχει αν υπάρχει και φορτώνει όλα τα αρχεία εικόνας (.png//.jpg) στον κατάλογο.
  - Παρουσιάζει τις κατηγορίες και το συνολικό πλήθος των εικόνων για λόγους debugging

def __len__(self):

- Επιστρέφει το μέγεθος του dataset, δηλαδή τον αριθμό των εικόνων που περιέχει το self.data.

def __getitem__(self, idx):img_path, label = self.data[idx]:

Ανακτά το μονοπάτι της εικόνας και την ετικέτα κατηγορίας από το self.data.

- Ελέγχει αν η ετικέτα είναι έγκυρη (π.χ., δεν ξεπερνά τον αριθμό των κατηγοριών).
- Φορτώνει την εικόνα από το αρχείο, τη μετατρέπει σε RGB και εφαρμόζει τους μετασχηματισμούς αν υπάρχουν.
- Επιστρέφει το ζεύγος (image, label) που θέλουμε

def display_batch(self, indexes):

Είσοδος:

- indexes: Μια λίστα με δείκτες εικόνων που θα εμφανιστούν.

Λειτουργία:

- Δημιουργεί ένα πλέγμα (5x5) για την εμφάνιση 25 εικόνων.
- Ανακτά τις εικόνες και τις ετικέτες χρησιμοποιώντας τη μέθοδο __getitem__.
- Εμφανίζει κάθε εικόνα με τίτλο την κατηγορία της (π.χ., "COVID", "Normal").
- Απενεργοποιεί τυχόν κενά υποπλαίσια αν η λίστα indexes έχει λιγότερα από 25 στοιχεία.
- Είναι χρήσιμη για την οπτική επιθεώρηση των δεδομένων και την ανίχνευση σφαλμάτων.
- _Εκτύπωση της καθε ετικέτας για λογους debugging (print(f"Index {idx}, Label: {label} ({self.classes[label]})")

Μετασχηματισμοί και κατασκευή του dataset(DATASET = COVID19Dataset(root_dir=path, transform=transformation))

- Ορίζονται μετασχηματισμοί:
    - Αλλαγή μεγέθους των εικόνων σε 128x128 pixels.
    - Μετατροπή τους σε Tensor (απαραίτητο για την χρήση στο PyTorch).
- Το dataset κατασκευάζεται με το COVID19Dataset, παρέχοντας το path δεδομένων και τους μετασχηματισμούς.

Παρουσίαση τυχαίων εικόνων

- Επιλέγονται 25 τυχαίοι δείκτες με την χρηση της random.sample.
- Η μέθοδος display_batch χρησιμοποιείται για την εμφάνιση των εικόνων.

Γράφημα κατανομής

- Εμφανίζεται το πλήθος των εικόνων ανά κατηγορία μέσω ραβδογράμματος, με τίτλους και τις αντιστοιχες ετικέτες της κάθε κατηγορίας στους άξονες

3 Συναρτήσεις εκπαίδευσης και δοκιμής

def confusion_matrix(y, y_pred, num_classes):

- **Είσοδοι:** Δέχεται τις πραγματικές κλάσεις (y), τις εκτιμώμενες κλάσεις (y_pred), και τον αριθμό των κλάσεων (num_classes)
- Δημιουργεί έναν πίνακα μηδενικών με διαστάσεις num_classes x num_classes
- Για κάθε ζεύγος πραγματικής και εκτιμώμενης κλάσης (y//y_pred) αυξάνει κατά 1 την αντίστοιχη θέση του πίνακα
- Και Επιστρέφει τον πίνακα σύγχυσης

def train_one_epoch(model, dataloader, optimizer, loss_fn, device):

Είσοδοι:Δέχεται το μοντέλο, έναν DataLoader με τα δεδομένα εκπαίδευσης, έναν optimizer, μία συνάρτηση απώλειας, και το device

- Θέτει το μοντέλο σε λειτουργία εκπαίδευσης (model.train()).
- Διατρέχει τα batches του DataLoader:
  - Μεταφέρει τα δεδομένα στην device.
  - Υπολογίζει την απώλεια για το batch.
  - Ενημερώνει τα βάρη μέσω του optimizer.
  - Ενημερώνει το loss και τις σωστές προβλέψεις).
- Υπολογίζει τη μέση απώλεια και την ακρίβεια.(avg_loss = running_loss / len(dataloader.dataset) accuracy = correct / total)

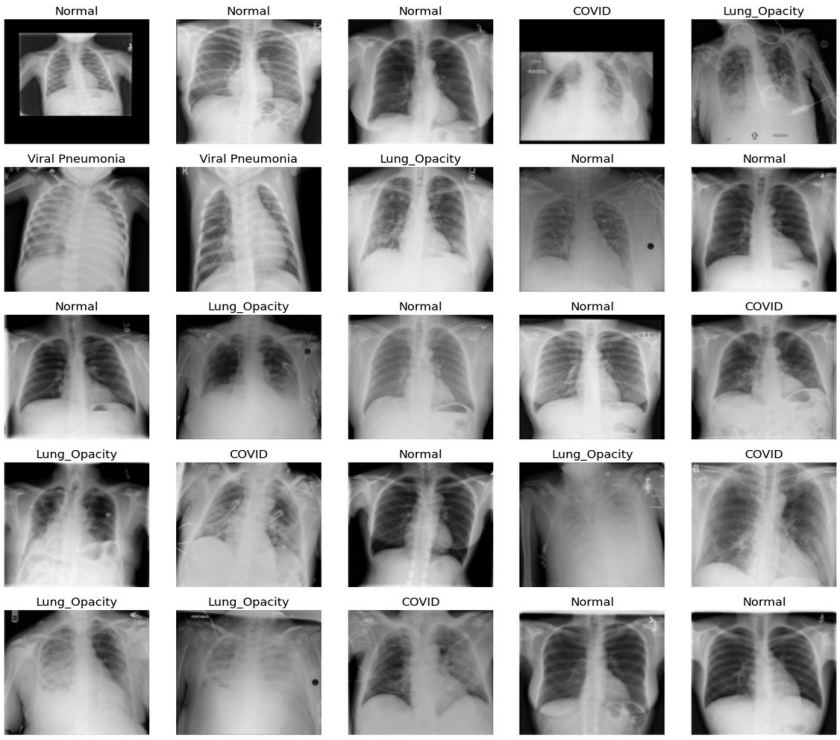
- Επιστρέφει τη μέση απώλεια και την ακρίβεια.

def test(model, dataloader, loss_fn, device, num_classes):

- **Είσοδοι:**Δέχεται το μοντέλο, έναν DataLoader με δεδομένα δοκιμής, μία συνάρτηση απώλειας, το device, και τον αριθμό κλάσεων.
- Θέτει το μοντέλο σε λειτουργία αξιολόγησης (model.eval()).
- Χρησιμοποιεί την torch.no_grad() για να απενεργοποιήσει τον υπολογισμό των gradients
- Διατρέχει τα batches του DataLoader
- Υπολογίζει την απώλεια.
- Υπολογίζει τις σωστές προβλέψεις και αποθηκεύει τις πραγματικές και εκτιμώμενες ετικέτες.
- Υπολογίζει τον πίνακα σύγχυσης με την κλήση της confusion_matrix που έχει ήδη υλοποιηθεί
- Επιστρέφει τη μέση απώλεια, την ακρίβεια, και τον confusion matrix

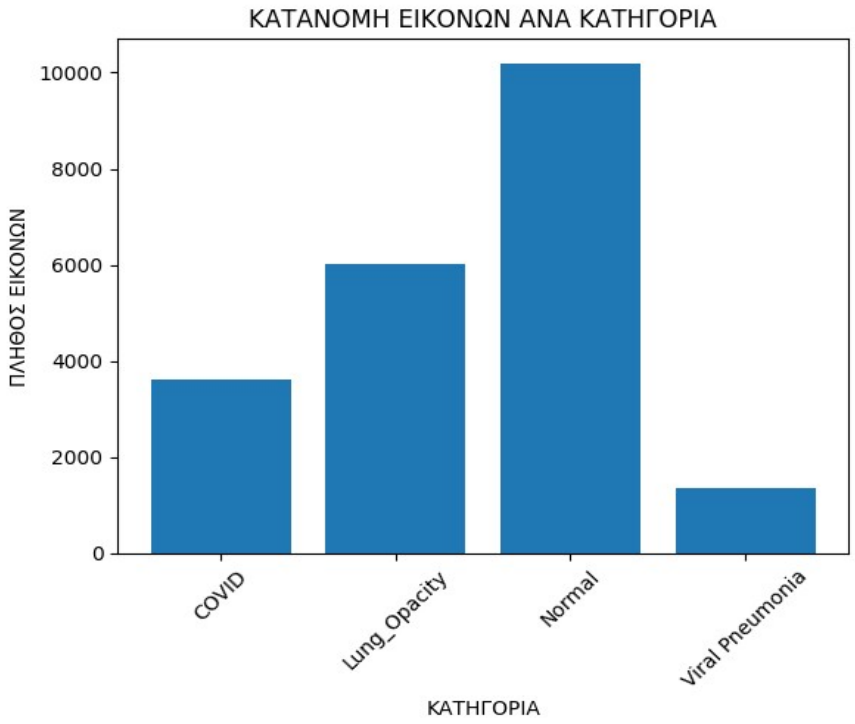
Ζητούμενο 1:

Λίστα με 25 τυχαίων indexes εικόνων του συνόλου δεδομένων



-

Πλήθος των εικόνων της κάθε κλάσης σε ραβδόγραμμα



Πλήθος εικόνων ανα κατηγορία:
COVID: 3616
Lung_Opacity: 6012
Normal: 10192
Viral Pneumonia: 1345

Σχολιασμός Αποτελεσμάτων:Παρατηρούμε ότι οι κατηγορίες Normal (10,192 εικόνες) και Lung_Opacity (6,012 εικόνες) έχουν τις περισσότερες εικόνες, ενώ οι κατηγορίες Covid (3,616 εικόνες) και Viral Pneumonia (1,345 εικόνες) έχουν λιγότερες σύμφωνα με τα τελικά αποτελέσματα .Αυτό επιβεβαιώνεται και από το ραβδόγραμμα που δείχνει την ανισοκατανομή των δεδομένων καθαρά δίνοντας εμφαση στισ διαφορές στις ποσότητες των εικόνων ανά κατηγορία. .Το ανησυχητικό στην περίπτωση μας ειναι οτι αυτη η ανισορροπία στις κατηγορίες (με μεγαλύτερο αριθμό στις φυσιολογικές ακτινογραφίες θώρακος) μπορεί να επηρεάσει τη συνολική απόδοση του μοντέλου που θα εκπαιδευτεί με τα συγκεκριμένα δεδομένα. Συγκεκριμένα το μοντέλο μπορεί να είναι προκατειλημμένο προς τις κατηγορίες με περισσότερα δεδομένα (Normal//Lung Opacity), οδηγώντας σε χαμηλότερη ακρίβεια πρόβλεψης για τις κατηγορίες με λιγότερα δείγματα (Viral Pneumonia, COVID).

4 Απλό συνελικτικό δίκτυο

Συνελικτικά Επίπεδα και Ενεργοποίηση

- **Conv2d:** Το πρώτο επίπεδο εφαρμόζει 8 φίλτρα 3×3 πάνω στα δεδομένα εισόδου (grayscale εικόνες με 1 κανάλι). Το δεύτερο επίπεδο επεξεργάζεται τα 8 κανάλια από το πρώτο επίπεδο και δημιουργεί 16 νέα χαρακτηριστικά.
- **ReLU:** Μετά από κάθε συνελικτική στρώση εφαρμόζεται η συνάρτηση ενεργοποίησης ReLU για να πετύχει την μη γραμμικότητα στο δίκτυο.

Επίπεδα Συγκέντρωσης

- **MaxPool2d:** Μειώνει τη χωρική διάσταση των χαρακτηριστικών με παράθυρο 2×2 και βήμα 2 όπως ζητήθηκε. Αυτό κάνει πιο γρήγορη την

εκπαίδευση και μειώνει το μέγεθος των δεδομένων (self.pool = nn.MaxPool2d(kernel_size=2, stride=2) #Εφαρμόζει ένα παράθυρο 2×2 kernel size)

Flatten

- Με τη μέθοδο view, μετατρέπουμε την έξοδο του τελευταίου συνελικτικού επιπέδου σε ένα μονοδιάστατο διάνυσμα πράγμα το οποίο χρειάζεται για τη σύνδεση με τα πλήρως συνδεδεμένα επίπεδα(x = x.view(x.size(0), -1))

Πλήρως Συνδεδεμένα Επίπεδα

- Το πρώτο επίπεδο έχει 32 νευρώνες, οι οποίοι επεξεργάζονται τα χαρακτηριστικά της εικόνας. Το δεύτερο επίπεδο εξόδου προβλέπει τις 4 πιθανές κατηγορίες.(COVID,Normal,Lung Opacity,Viral Pneumonia)

Μετασχηματισμοί Εικόνων

- Grayscale: Οι εικόνες μετατρέπονται σε ασπρόμαυρες (1 κανάλι) για απλοποίηση(transforms.Grayscale(num_output_channels=1))
- Resize: Οι εικόνες προσαρμόζονται σε διαστάσεις 28×28 για συμβατότητα με το μοντέλο. transforms.Resize((28, 28))
- ToTensor: Μετατροπή των εικόνων σε tensors, τη μορφή που απαιτεί η PyTorch (transforms.ToTensor())
- Normalize: Κανονικοποίηση τιμών ώστε τα δεδομένα να βρίσκονται στο διάστημα [−1,1], βελτιώνοντας τη σταθερότητα της εκπαίδευσης (transforms.Normalize(mean=[0.5], std=[0.5]))

Διαχωρισμός Συνόλων Δεδομένων

Με την random_split, τα δεδομένα χωρίζονται όπως ζητήθηκε σε:

- 60% για εκπαίδευση
- 20% για επικύρωση
- 20% για δοκιμή

*Το seed (42) εξασφαλίζει ότι τα αποτελέσματα είναι αναπαραγώγιμα.

Ρυθμίσεις Εκπαίδευσης

- Optimizer: Adam με lr=0.001.
- Loss Function: CrossEntropyLoss για ταξινόμηση πολλών κλάσεων.
- Early Stopping: Ελέγχεται η απόδοση στο validation set, και σταματάει η εκπαίδευση αν η απώλεια δεν βελτιώνεται για 5 συνεχόμενες epochs συμφωνα με τις προηγούμενες υλοποιήσεις

Εκπαίδευση

- Loop Εκπαίδευσης
    - Εκτελείται μια εποχή εκπαίδευσης στο training set
    - Αξιολογείται η απόδοση στο validation set
    - *Αποθηκεύεται το καλύτερο μοντέλο.*

Επικύρωση

- Αξιολόγηση σε validation data: Υπολογίζονται απώλεια, ακρίβεια και πίνακας σύγχυσης για τα δεδομένα επικύρωσης.
- Αποθήκευση καλύτερου μοντέλου: Αν η απώλεια επικύρωσης βελτιωθεί, αποθηκεύεται το τρέχον μοντέλο.
- Early Stopping

Τελικό Τεστ

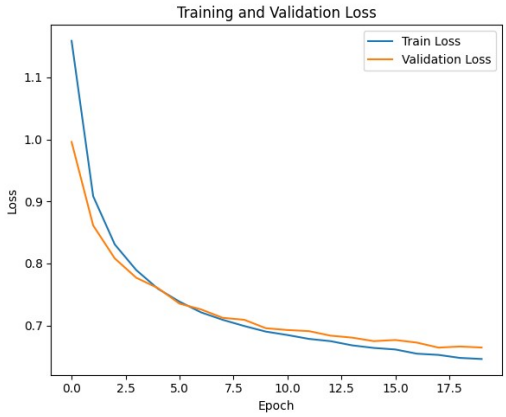
Αξιολογείται το αποθηκευμένο καλύτερο μοντέλο και υπολογίζονται:

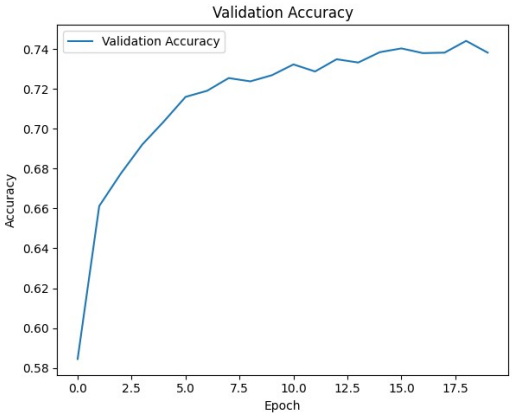
- Η απώλεια δοκιμής (test loss).
- Η ακρίβεια δοκιμής (test accuracy).
- Ο confusion matrix για ανάλυση λανθασμένων προβλέψεων.

Οπτικοποίηση Αποτελεσμάτων (για λόγους debugging)

Τα γραφήματα παρέχουν:

- Την πορεία της απώλειας εκπαίδευσης // επικύρωσης και τνν πορεία της ακρίβειας επικύρωσης





Ζητούμενο 2 :

Ευστοχία του Μοντέλου (Accuracy)

Από τον κώδικα και την αξιολόγηση του μοντέλου σε κάθε σύνολο δεδομένων:

- **Σύνολο Εκπαίδευσης:**ξεκινά από περίπου 45.55% και αυξάνεται σταθερά έως περιπου 75.06% στο τέλος της εκπαίδευσης
- **Σύνολο Επικύρωσης:**ξεκινά από 58.45% και σταθεροποιείται περίπου στο 73.82%
- **Σύνολο Δοκιμής:** φτάνει το 73.87% στο τέλος, υποδεικνύοντας ότι το μοντέλο διατηρεί μια σχετικά καλή γενίκευση στα δεδομένα δοκιμής

```
Epoch 1/20
Train Loss: 1.1587, Train Accuracy: 45.55%
Val Loss: 0.9959, Val Accuracy: 58.45%
Confusion Matrix:
tensor([[  15,  109,  567,    0],
        [  12,  504,  689,    0],
        [   0,   81, 1955,    0],
        [   5,   22,  274,    0]])
```

```
Epoch 20/20
Train Loss: 0.6461, Train Accuracy: 75.06%
Val Loss: 0.6646, Val Accuracy: 73.82%
Confusion Matrix:
tensor([[ 336,  169,  178,    8],
        [  88,  879,  228,   10],
        [ 138,  178, 1673,   47],
        [  15,   20,   29,  237]])
```

```
Final Test Results:
Test Loss: 0.6669, Test Accuracy: 73.87%
Confusion Matrix:
tensor([[ 347,  147,  238,    6],
        [ 100,  831,  262,   11],
        [ 110,  132, 1759,   38],
        [  16,   16,   30,  190]])
```

Δεδομένης της κατανομής των κατηγοριών στο σύνολο δεδομένων, εξετάσαμε λίγο πιο λεπτομερώς τι συμβαίνει στο μοντέλο μας

Δεδομένης της κατανομής των κατηγοριών στο σύνολο δεδομένων, το μοντέλο από ότι βλέπουμε παρουσιάζει σημάδια ανισησ κατανομής στις προβλέψεις, με τις κατηγορίες που εχουν μεγαλύτερο αριθμό να κυριαρχούν στις σωστέσ ταξινομήσεις. Ο πίνακας σύγχυσης υποδεικνύει ότι οι κατηγορίες με λιγότερα δείγματα εμφανίζουν αυξημένα ποσοστά εσφαλμένων προβλέψεων

Ανισορροπία Κατηγοριών και Σύγχυση

- Η ανισορροπία που παρουσιαζεται στις κατηγορίες όπως προαναφέραμε μπορεί να οδηγήσει το μοντέλο να είναι πιο "ευαίσθητο" στις πιο συχνές κατηγορίες (π.χ. Normal//Lung Opacity) και να παρουσιάζει υψηλότερη ακρίβεια στην ταξινόμηση αυτών των συγκεκριμένων κατηγοριών. Αυτό εξηγεί γιατί η κατηγορία Normal έχει πολύ καλύτερη απόδοση (π.χ., 1759 από 1828 σωστά ταξινομημένα δείγματα στην τελική φάση//test set)

<u>Ανάλυση της Επίδοσης σε Κάθε Κλάση με Βάση τον Πίνακα Σύγχυσης</u>

<u>COVID</u>

- το μοντέλο μπερδεύει περισσότερο τις εικόνες COVID με την κατηγορία Normal(Τα 238 δείγματα από την κατηγορία Normal ταξινομήθηκαν λανθασμένα ως COVID) με αποτέλεσμα να επιβεβαιώνεται ότι το μοντέλο έχει καλύτερη απόδοση στην κατηγορία Normal, αλλά με κάποια σύγχυση με την κατηγορία COVID (ιδιαίτερα σε περιπτώσεις που τα χαρακτηριστικά των εικόνων του COVID μοιάζουν με φυσιολογικές εικόνες).
- παρατηρούμε επίσης ότι το μοντέλο έχει 147 λανθασμένα ταξινομημένα δείγματα ως Lung Opacity. Αυτό δείχνει ότι μπέρδευει τις εικόνες COVID με εκείνες που ανήκουν στην κατηγορία Lung Opacity αυτό ειναι ιδιαίτερα σημάντικο διότι το μοντέλο έχει συνολικά πολύ περισσότερες σωστές ταξινομήσεις στην κατηγορία Normal, γεγονός που κάνει τα λάθη να φαίνονται ποσοστιαία μικρότερα και λιγότερο σημαντικά σε σχέση με την

κατηγορία COVID, όπου ο αριθμός των δειγμάτων είναι μικρότερος.

Lung Opacity

Η Lung Opacity είναι η δεύτερη πιο συχνή κατηγορία (6012 εικόνες) και φαίνεται να έχει σχετικά καλή απόδοση στο μοντέλο συγκεκριμένα παρατηρούμε οτι

- Υπάρχει κάποια σύγχυση με την κατηγορία Normal (262 δείγματα) και σε αυτήν την περίπτωση ,ενδεχωμένος γιατι οι εικόνες Lung Opacity σε κάποιες περιπτώσεις μπορεί να μοιάζουν περισσότερο με εικόνες φυσιολογικών πνευμόνων
- Επίσης, με την κατηγορία COVID υπάρχει μια μικρή σύγχυση οπου και σε αυτήν την περίπτωση είναι σημαντική διότι το μοντέλο έχει συνολικά πολύ περισσότερες σωστές ταξινομήσεις στην κατηγορία Normal, γεγονός που κάνει τα λάθη να φαίνονται ποσοστιαία μικρότερα και λιγότερο σημαντικά σε σχέση με την κατηγορία COVID, όπου ο αριθμός των δειγμάτων είναι μικρότερος.

Normal

- Η κατηγορία Normal, λόγω της υψηλής της συχνότητας στο σύνολο δεδομένων (10192 εικόνες), έχει πολύ καλή απόδοση.
    - Επειδή η κατηγορία Normal περιλαμβάνει φυσιολογικές εικόνες χωρίς σημαντικές αλλοιώσεις, είναι πιο εύκολο για το μοντέλο να την αναγνωρίσει σωστά.
    - Εισησ παρατηρείται οτι μερικέσ φόρεσ ,μπερδέυεται με την κατηγορία Lung Opacity πεισσότερο σε ορισμένα δείγματα

Viral Pneumonia

- Η κατηγορία Viral Pneumonia είναι η λιγότερο εκπροσωπούμενη, με μόλις 1345 εικόνες, γεγονός που κάνει το μοντέλο λιγότερο ακριβές στην ταξινόμησή της.
    - Στην Confusion Matrix, παρατηρούμε αρκετές λάθες ταξινομήσεις με την κατηγορία Normal.

- Επιπλέον, υπάρχει κάποια σύγχυση και με την κατηγορία Lung Opacitykαι COVID, καθώς μπορεί να υπάρχουν κοινά χαρακτηριστικά σε εικόνες που αφορούν λοιμώξεις των πνευμόνων.

Σημασία των* αποτελεσμάτων *μας στην κλινική πράξη

Η ισορροπημένη απόδοση στις κατηγορίες παρά την ανισομερή κατανομή των εικόνων (π.χ., 10.192 Normal και μόνο 1.345 Viral Pneumonia) δείχνει ότι το μοντέλο μαθαίνει και γενικεύει καλά σε γενικές γραμμές. Ωστόσο, τα αποτελέσματα δείχνουν ότι:

- Το μοντέλο ενδέχεται να επηρεάζεται από τη μεγαλύτερη εκπροσώπηση της κατηγορίας Normal.
- Η σύγχυση μεταξύ COVID και Lung Opacity μπορεί να οφείλεται στο γεγόνος οτι δεν γινονταί αντιλιπτέσσ μερικές λεπτομερειεσ για την διαφοροποίηση των 2 ασθενειών.

Σε γενικές γραμμές επομένως τα αποτελέσματα μας στην κλινική πράξη μπόρουν βοηθήσουν στους εξής τομείς:

<u>Υποστήριξη Διαγνωστικής Απόφασης</u>

Το μοντέλο μπορεί να λειτουργήσει ως εργαλείο βοηθείας για τους κλινικούς γιατρούς δηλαδη

- Ανίχνευση πιθανών περιπτώσεων COVID-19: Μπορεί να εντοπίζει ύποπτες περιπτώσεις που απαιτούν περαιτέρω κλινική αξιολόγηση.
- Ταξινόμηση άλλων πνευμονικών παθήσεων**:** Η διάκριση μεταξύ φυσιολογικών εικόνων και εικόνων που δείχνουν Lung Opacity ή Viral Pneumonia μπορεί να υποδείξει την ύπαρξη λοιμώξεων, όπως πνευμονία ή άλλες πνευμονικές βλάβες

<u>Χρονική Αποδοτικότητα και Μείωση Φορτίου Εργασίας</u>

- Μπορεί να επιταχύνει τη διαδικασία διάγνωσης σε περιπτώσεις αυξημένου φόρτου εργασίας για τους ακτινολόγους συγκεκριμένα
- Βοηθά στην έγκαιρη διάγνωση πνευμονικών λοιμώξεων σε συνθήκες όπου η πρόσβαση σε εξειδικευμένο προσωπικό είναι περιορισμένη

<u>Εφαρμογές σε Πανδημικές Καταστάσεις</u>

Σε περιπτώσεις πανδημίας, όπως με την COVID-19, η ταχύτητα και η ακρίβεια ανίχνευσης είναι κρίσιμη:

- Το μοντέλο μπορεί να χρησιμοποιηθεί σε περιβάλλοντα μαζικού ελέγχου για γρήγορη προεπιλογή ασθενών που απαιτούν πιο εκτεταμένες κλινικές εξετάσεις.

Περιορισμοί και Προκλήσεις

Παρόλο που τα αποτελέσματα είναι πολύ βοηθητικά ***δυτυχώς υπάρχουνμ καποιοι σημαντικοί περιορισμοί:***

- Σύγχυση μεταξύ κατηγοριών: Η κατηγοριοποίηση των COVID και Lung Opacity χρειάζεται περαιτέρω βελτίωση, καθώς η αρκετή σύγχυση τουςμπορεί να οδηγήσει σε λάθη διάγνωσης
- Ανισομερής κατανομή κατηγοριών: Η μικρή εκπροσώπηση της κατηγορίας Viral Pneumonia μπορεί να επηρεάζει αρνητικά την απόδοση.
- Γενίκευση: Η αποτελεσματικότητα του μοντέλου σε πραγματικά κλινικά δεδομένα που προέρχονται από διαφορετικούς πληθυσμούς και συνθήκες πρέπει να αξιολογηθεί.

Συμπέρασμα:συμπερασματικά, το μοντέλο έχει τη δυνατότητα να συμβάλει θετικά στη βελτίωση της διαγνωστικής διαδικασίας, μειώνοντας τον χρόνο ανάλυσης εικόνων και ενισχύοντας την ακρίβει***α. Ωστόσο, απαιτείται προσεκτική αξιολόγηση και συνεχιζόμενη βελτίωση πριν από την πλήρη κλινική του εφαρμογή για να αποφεχθούν οι λάθοσ διαγνωσεις καθως ύπαρχουν αν οχι πολλά αρκέτα "λάθη" στο μοντελο μας που πρέπει να ληφθούν υπόψην για την αποφύγη τραγικών λαθών σε τόσο σημαντικές εξετάσεις στον τομέα της υγείας***

5 Συνελικτικό δίκτυο μεγαλύτερου βάθους

Ορισμός του Μοντέλου

- Συνελικτικά Επίπεδα (Conv Layers): 10 συνολικά συνελικτικά επίπεδα που χρησιμοποιούν φίλτρα 3x3 με padding=1 για διατήρηση των διαστάσεων.

- ReLU Ενεργοποίηση: Μεταξύ των συνελικτικών επιπέδων για μη γραμμικότητα.
- Max Pooling Layers: Για μείωση της διαστατικότητας με strides 4 και 2.
- Fully Connected Layers: Ένα πλήρως συνδεδεμένο επίπεδο υπολογίζεται δυναμικά για τις εισόδους από το τελευταίο επίπεδο.
- Αριθμός Κλάσεων: Ορίζεται από το dataset.

Προετοιμασία Δεδομένων

- Dataset: Χρησιμοποιεί το COVID19Dataset για δεδομένα (DATASET = COVID19Dataset(root_dir= '/content/COVID19Dataset', transform=transformation))
- Μετασχηματισμοί (Transformations):
    - Μετατροπή εικόνων σε grayscale.
    - Αλλαγή μεγέθους εικόνων σε 128x128.
    - Μετατροπή σε tensors.
- Διαχωρισμός: Το dataset χωρίζεται σε training, validation και test sets (60%-20%-20%) όπως ορίζεται στα προηγούμενα συνελικτίκα δίκτυα

Εκπαίδευση

- Loop Εκπαίδευσης
    - Εκτελείται μια εποχή εκπαίδευσης στο training set.
    - Αξιολογείται η απόδοση στο validation set.
    - Αποθηκεύεται το καλύτερο μοντέλο.

Μέτρηση Απόδοσης:

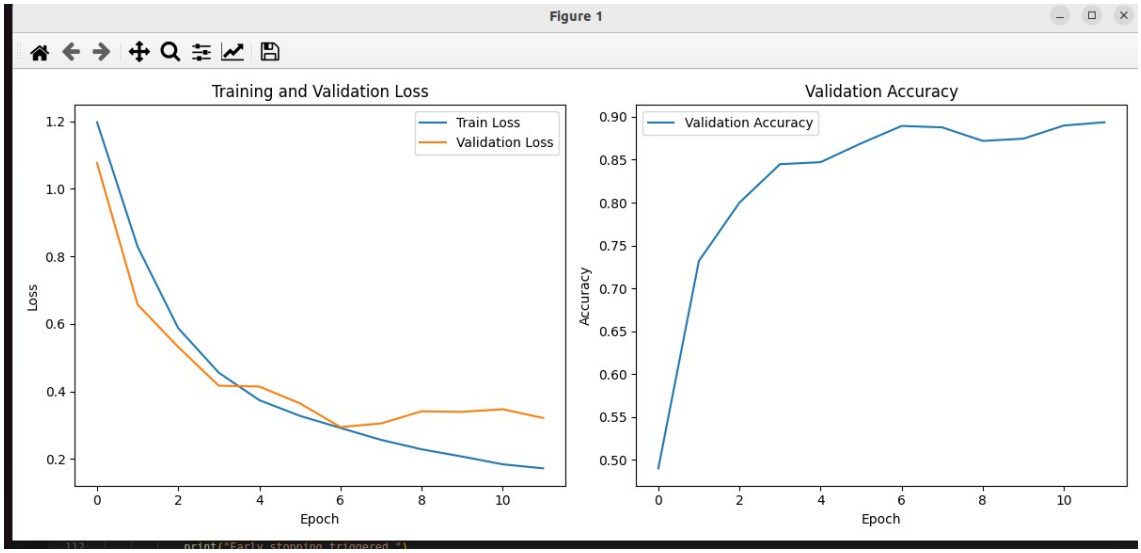
- Υπολογίζεται η απώλεια και η ακρίβεια για training και validation

Αξιολόγηση

- Το καλύτερο μοντέλο φορτώνεται και αξιολογείται στο test set
- Υπολογίζεται η απώλεια, η ακρίβεια και ο confusion matrix.

Γραφήματα

- Σχεδιάζονται γραφήματα για την εξέλιξη της απώλειας και της ακρίβειας κατά τη διάρκεια της εκπαίδευσης για λόγους debugging



Ζητούμενο 3:

```
Epoch 1/20
Train Loss: 1.1625, Train Accuracy: 50.37%
Val Loss: 0.9705, Val Accuracy: 59.44%
Confusion Matrix:
tensor([[   0,  296,  389,    6],
        [   0,  896,  309,    0],
        [   0,  531, 1469,   36],
        [   0,   31,  119,  151]])
```

```
Epoch 16/20
Train Loss: 0.1137, Train Accuracy: 95.79%
Val Loss: 0.3629, Val Accuracy: 89.42%
Confusion Matrix:
tensor([[ 613,   48,   27,    3],
        [  14, 1056,  134,    1],
        [  27,  153, 1844,   12],
        [   8,    2,   19,  272]])
Early stopping triggered.
```

```
Final Test Results:
Test Loss: 0.3065, Test Accuracy: 88.68%
Confusion Matrix:
tensor([[ 698,   27,   11,    2],
        [  41,  972,  188,    3],
        [  86,   97, 1849,    7],
        [   3,    0,   14,  235]])
```

Train Accuracy//Validation Accuracy

- **CNN1:**
  - Train Accuracy:75.06%
  - Validation Accuracy: 73,82%

- **CNN2:**
    - Train Accuracy: 95.79%
    - Validation Accuracy: 89.42%

Το CNN2 παρουσιάζει σαφώς καλύτερη επίδοση κατά την εκπαίδευση και την επικύρωση.

Validation loss//Train loss

- **CNN1:**
    - Train loss: 0.6461
    - Validation loss: 0.6646
- **CNN2:**
    - Train loss: 0.1137
    - Validation loss: 0.3629

Η μείωση της απώλειας στο CNN2 υποδεικνύει καλύτερη γενίκευση και αποτελεσματικότερη προσαρμογή στο σύνολο δεδομένων

Τελική Απόδοση Δοκιμαστικού Συνόλου

- **CNN1:**
    - Test Accuracy: 73.87%
    - Test loss: 0.6669
- **CNN2:**
    - Test Accuracyς: 88.68%
    - Test loss: 0.3065

Η σύγκριση των πινάκων σύγχυσης για τα δύο μοντέλα CNN1 και CNN2 παρουσιάζει μια λεπτομερή εικόνα της απόδοσης των μοντέλων για κάθε κατηγορία

Ανάλυση Ανά Κατηγορία

1. **Κατηγορία 0**
    - CNN2: 698 σωστές ταξινομήσεις (TP), 40 λανθασμένες ταξινομήσεις (FN).

- CNN1: 347 σωστές ταξινομήσεις (TP), 391 λανθασμένες ταξινομήσεις (FN).
- Το CNN2 υπερτερεί σαφώς στην ακριβή αναγνώριση της κατηγορίας 0 με λιγότερες λανθασμένες ταξινομήσεις.

2. **Κατηγορία 1**
    - CNN2: 972 σωστές (TP), 232 λανθασμένες (FN).
    - CNN1: 831 σωστές (TP), 373 λανθασμένες (FN).
    - Το CNN2 παρουσιάζει καλύτερη ακρίβεια στην κατηγορία 1, αν και η διαφορά είναι μικρότερη σε σχέση με την κατηγορία 0.
3. **Κατηγορία 2**
    - CNN2: 1849 σωστές (TP), 190 λανθασμένες (FN).
    - CNN1: 1759 σωστές (TP), 280 λανθασμένες (FN).
    - Η κατηγορία αυτή δείχνει επίσης σημαντική βελτίωση στο CNN2.
4. **Κατηγορία 3**
    - CNN2: 235 σωστές (TP), 17 λανθασμένες (FN).
    - CNN1: 190 σωστές (TP), 62 λανθασμένες (FN).
    - Το CNN2 σημειώνει εντυπωσιακά λιγότερες λανθασμένες ταξινομήσεις στην κατηγορία 3.

Η διαφορά στην απόδοση μεταξύ του μοντέλου CNN1 και του CNN2 οφείλεται σε διάφορους παράγοντες που σχετίζονται με την αρχιτεκτονική και την εκπαίδευση των δικτύων:

1. **Αρχιτεκτονική του Δικτύου**

CNN1:

- Απλούστερη δομή με δύο συνελικτικά επίπεδα και περιορισμένο αριθμό φίλτρων (8 και 16 φίλτρα)
- Χρήση ενός μόνο μικρου πυρήνα συγκέντρωσης (Max Poolingg με βήμα 2)
- Τα δεδομένα κανονικοποιούνται σε 28x28 διαστάσεις, γεγονός που περιορίζει τη χωρητικότητα του δικτύου να ανιχνεύσει λεπτομέρειες.

CNN2:

- Πολύπλοκη αρχιτεκτονική με πολλαπλά συνελικτικά επίπεδα (10 επίπεδα συνολικά).

- Πολλαπλά στάδια Max Pooling και αρκετά μεγάλη αύξηση στον αριθμό φίλτρων (από 32 έως 512).
- Δυνατότητα εντοπισμού σύνθετων χαρακτηριστικών λόγω μεγαλύτερης χωρητικότητας του μοντέλου.
- Είσοδοι εικόνων μεγαλύτερης ανάλυσης (128x128) που αφομειωνουν περισσότερες λεπτομέρειες.

2. Μέγεθος του Μοντέλου

Το CNN2 έχει μεγαλύτερο αριθμό παραμέτρων, επιτρέποντάς του να εκπαιδεύεται σε πιο σύνθετες καταστάσεις και να μαθαίνει περισσότερα χαρακτηριστικά. Αυτό οδηγεί σε:

- Υψηλότερη ακρίβεια στην ταξινόμηση.
- Καλύτερη γενίκευση λόγω της αυξημένης χωρητικότητας για την αναπαράσταση των δεδομένων.

3. *Διαχείριση Πληροφοριών*

Το **CNN2** μπορεί να διατηρήσει χωρικές πληροφορίες καλύτερα με την βοήθεια των συνελικτικών επιπέδων με μεγαλύτερα poolings. Αυτό επιτρέπει την αποδοτικότερη ανάλυση των εικόνων σε διαφορετικές κλίμακες και βοηθάει στη βελτιωμένη ταξινόμηση.

Συμπέρασμα

Η διαφορά στην απόδοση οφείλεται κυρίως στη βελτιωμένη αρχιτεκτονική, στην αυξημένη χωρητικότητα του CNN2, και στη δυνατότητα που έχοει το μοντέλο να ανάλυει δεδομένα μεγαλύτερης ανάλυσης. Ενώ ταυτόχρονα από την αλλη πλευρά η πολυπλοκότητα του CNN2 μπορεί να το καταστήσει ευάλωτο σε υπερπροσαρμογή σε μικρότερα σύνολα δεδομένων

6 Με χρήση προεκπαιδευμένου δικτύου

Προεπεξεργασία Δεδομένων:

- Μετασχηματισμός εικόνων σε κλίμακα του γκρι με 3 κανάλια//αλλαγή μεγέθους (224x224), μετατροπή σε tensors και κανονικοποίηση.

- Διαχωρισμός dataset σε training (60%), validation (20%) και test (20%) sets με σταθερό seed για αναπαραγωγιμότητα όπως στα προηγούμενα μοντέλα

Προετοιμασία Μοντέλου:

- Χρήση προεκπαιδευμένου ResNet50.
- Αντικατάσταση του τελικού fully-connected επιπέδου με νέο επίπεδο για ταξινόμηση σε αριθμό κλάσεων ίσο με το dataset.

Εκπαίδευση με Fine-Tuning:

- Εκπαίδευση ολόκληρου του δικτύου με learning rate 1×10−4, optimizer Adam, και loss function CrossEntropyLoss όπως προηγουμένως
- early stopping εάν η απόδοση του validation loss δεν βελτιωθεί για 5 εποχές.
- Αποθήκευση του καλύτερου μοντέλου βάσει του validation loss.

Αξιολόγηση Μοντέλου:

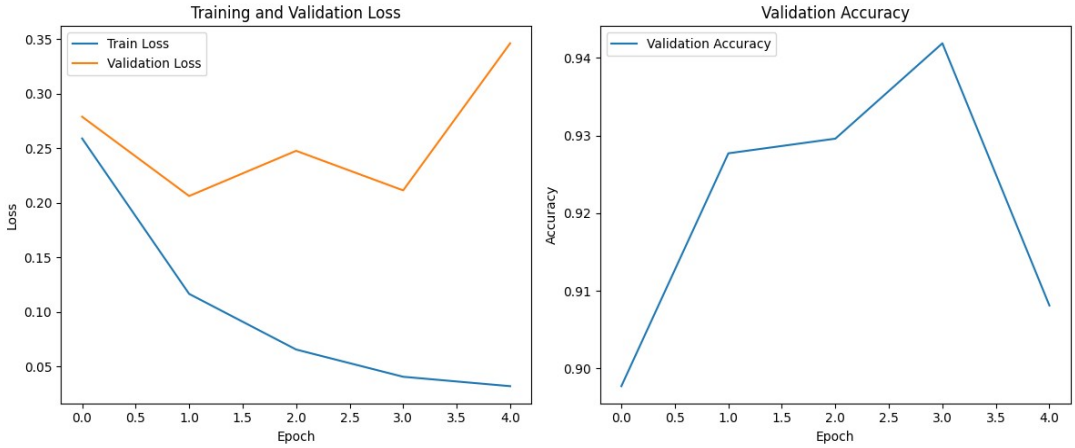
- Υπολογισμός του τελικού test loss, test accuracy και του confusion matrix.

Χρήση ResNet ως Εξαγωγέα Χαρακτηριστικών:

- Πάγωμα όλων των συνελικτικών επιπέδων του ResNet50.
- Επανεκπαίδευση μόνο του τελικού πλήρως συνδεδεμένου επιπέδου.
- Παρόμοια διαδικασία εκπαίδευσης και αξιολόγησης με fine-tuning.

Γραφήματα:

- Σχεδίαση γραφημάτων για την απώλεια (loss) και την ακρίβεια (accuracy) κατά τη διάρκεια της εκπαίδευσης για debugging.

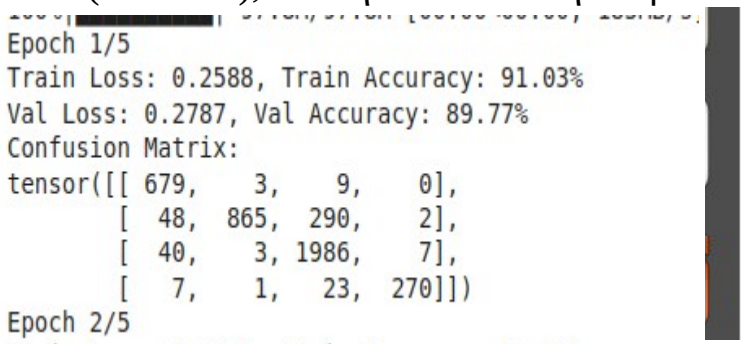


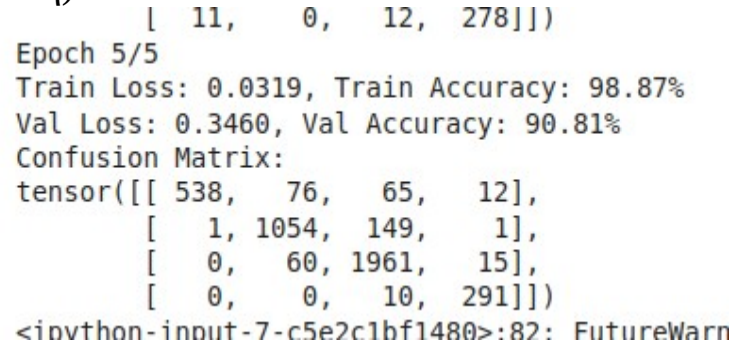
Ζητούμενο 4:

Το αρχικό μοντέλο ResNet50 παρουσιάζει αρκέτα καλη βελτίωση στην ακρίβεια εκπαίδευσης, φτάνοντας το 98.87%, ενώ η ακρίβεια επικύρωσης κορυφώνεται στο 94.19% στο τέταρτο epoch. Ομως η πτώση της απόδοσης επικύρωσης στο τελευταίο epoch, σε συνδυασμό με την αύξηση του Val Loss δείχνει μία ενδεχόμενη υπερεκπαίδευση. Το τελικό αποτέλεσμα στο test set (94.00% ακρίβεια) είναι πόλυ ικανοποιητικό,και δέιχνει οτι υπάρχει καλή γενίκευση, αλλά οι τασεις της απώλειας επικύρωσης χρειάζονται περισσότερη μελέτη για να σταθεροποίηθει η απόδοση σε ένα σημειο

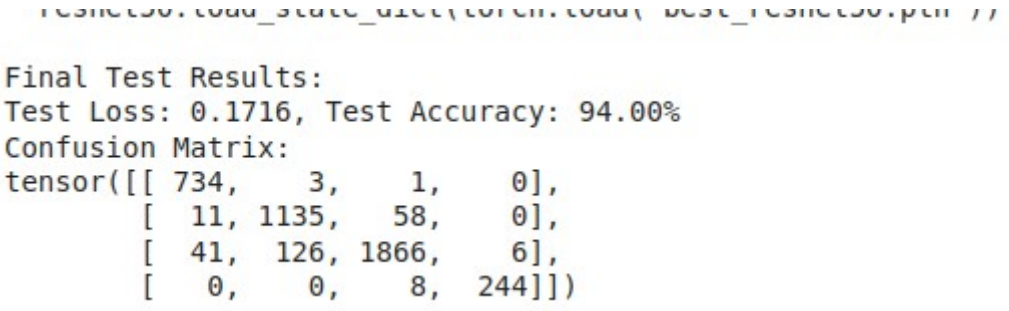
Κατά την εκπαίδευση ως εξαγωγέας χαρακτηριστικών, το ResNet50 παρουσίασε υψηλή και σταθερήTrain Accuracy 98.61% και χαμηλές τιμές απώλειας στην εκπαίδευση, πράγμα το οποίο οφειλεται στο γεγονος οτι παγώνουμε τα πρώτα στρωματα του μοντέλου και είναι λογίκο. Η ακρίβεια επικύρωσης έφτασε στο 95.11%, ξεπερνώντας την αρχική εκπαίδευση, ενώ οι τιμές της απώλειας επικύρωσης παρέμειναν χαμηλές και σταθερές. Στο test set, η ακρίβεια πήγε στο 95.42% με σημαντικά χαμηλότερη απώλεια (0.1319), επιδεικνύοντας καλύτερη γενίκευση και μικρότερη απόκλιση μεταξύ train και test, καθιστώντας το μοντέλο πιο αξιόπιστο για πραγματική χρήση.

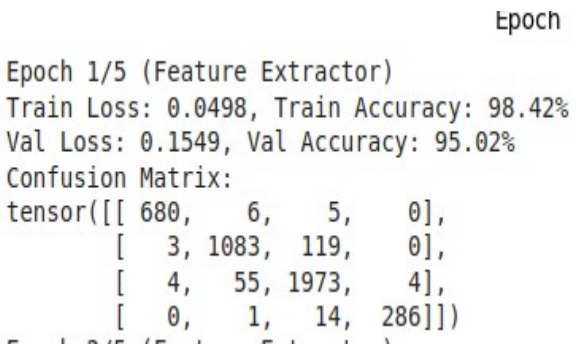
Συμπέρασμα :Η δέυτερη προσέγγιση του εξαγωγέα χαρακτηριστικών φαίνεται πιο αποτελεσματική για αυτό το συγκεκριμένο πρόβλημα, καθώς δίνει λιγο υψηλότερη απόδοση και καλύτερη σταθερότητα χωρίς σημάδια υπερεκπαίδευσης σε αντιθεση με την πρώτη προσέγγιση (η πτώση της απόδοσης επικύρωσης στο τελευταίο epoch μαζι με την αύξηση της απώλειας (Val Loss), υποδηλώνει πιθανή υπερεκπαίδευση)

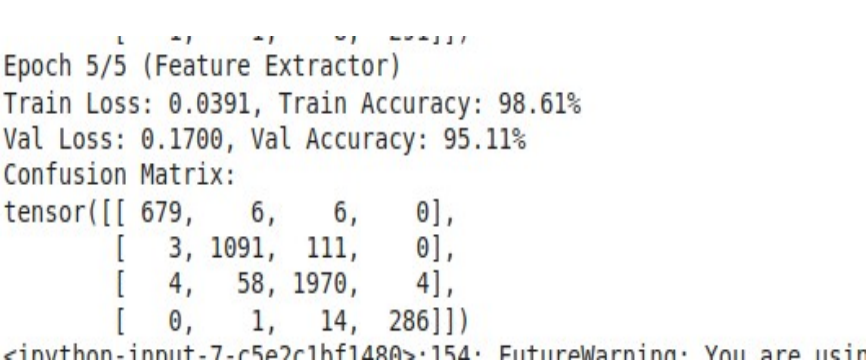


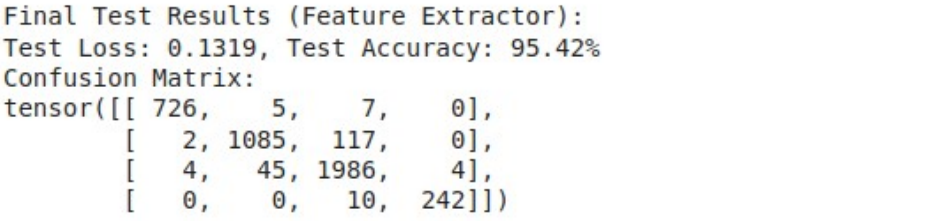


—









7.1 Μονάδα με παραλειπόμενες συνδέσεις

Δομή Μοντέλου:

Το μοντέλο περιλαμβάνει τέσσερα επίπεδα συνελικτικών μπλοκ με αριθμούς φίλυρων που επιλέξαμε : 64, 128, 256, 512.
Κάθε μπλοκ αποτελείται από δύο συνελικτικά επίπεδα με πυρήνα 3×3 και χρήση της ReLU
Περιλαμβάνει σύνδεση skip σε κάθε μπλοκ για αποφυγή εξαφάνισης των gradients.

Υπερπαράμετροι:

Αριθμός κατηγοριών: 4 (COVID, Lung Opacity, Normal, Viral Pneumonia)
Epochs: 5.
batch size: 64.
Ρυθμός μάθησης: 1×10−4.
Βελτιστοποιητής: Adam με β1=0.9, β2=0.99 ‘οπως χρησιμοποιήσαμε προηγουμενως
Συνάρτηση απώλειας: CrossEntropyLoss.

Μετασχηματισμοί Δεδομένων:

Μετατροπή εικόνας σε 3 κανάλια (RGB).
Αλλαγή μεγέθους εικόνας σε 224×224.
Κανονικοποίηση τιμών με μέσα [0.485, 0.456, 0.406] και τυπικές αποκλίσεις [0.229, 0.224, 0.225].

Δεδομένα:

Διαχωρισμός συνόλων: 60% για εκπαίδευση, 20% για επικύρωση, 20% για δοκιμή όπως και προηγουμένως
batch size: 64.

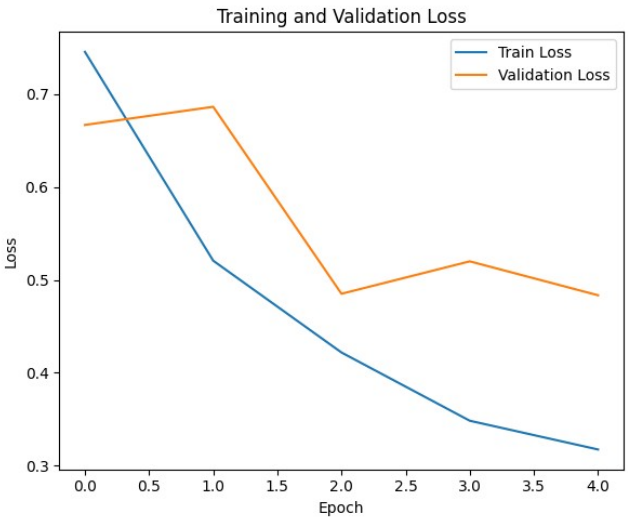
Εκπαίδευση: Το μοντέλο φτάνει σε Train Accuracy: 88.66% στο τέλος των εποχών.
Επικύρωση: Val Accuracy: 80.94%.
Τελική Δοκιμή: Test Accuracy: 81.48%.
Παρατηρείται σταθερή βελτίωση κατά τη διάρκεια εκπαίδευσης, με το καλύτερο μοντέλο να σώζεται μέσω early stopping.

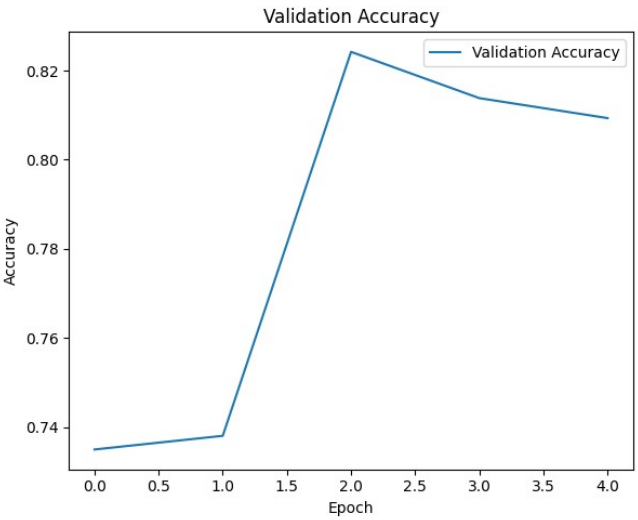
Συμπέρασμα: Το μοντέλο δείχνει ικανοποιητική ακρίβεια στην ταξινόμηση ακτινογραφιών θώρακα, επιδεικνύοντας αξιοπιστία στη διάκριση διαφορετικών κατηγοριών παθήσεων. Υπάρχει περιθώριο βελτίωσης, π.χ., με πιο σύνθετα μοντέλα ή διαφορετική ρύθμιση υπερπαραμέτρων

```
ΚΑΤΗΓΟΡΙΕΣ: ['COVID', 'Lung_Opacity', 'Normal', 'Viral Pneumonia']
ΣΥΝΟΛΟ ΕΙΚΟΝΩΝ: 21165
Epoch 1/5
Train Loss: 0.7452, Train Accuracy: 71.46%
Val Loss: 0.6666, Val Accuracy: 73.49%
Confusion Matrix:
tensor([[ 389,  135,  167,    0],
        [  31,  963,  211,    0],
        [ 137,  291, 1607,    1],
        [  17,   53,   79,  152]])
```

```
Epoch 5/5
Train Loss: 0.3176, Train Accuracy: 88.66%
Val Loss: 0.4835, Val Accuracy: 80.94%
Confusion Matrix:
tensor([[ 660,   27,    2,    2],
        [  39, 1108,   57,    1],
        [ 117,  433, 1367,  119],
        [   7,    3,    0,  291]])
<ipython-input-4-efb22217d7a2>:137: FutureWarning: You
```

```
Final Test Results:
Test Loss: 0.4595, Test Accuracy: 81.48%
Confusion Matrix:
tensor([[ 699,   34,    2,    3],
        [  31, 1126,   46,    1],
        [ 127,  406, 1379,  127],
        [   4,    3,    0,  245]])
```


Training and Validation Loss
Train Loss
Validation Loss
Loss
Epoch
0.3
0.4
0.5
0.6
0.7
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


Validation Accuracy
Validation Accuracy
Accuracy
Epoch
0.74
0.76
0.78
0.80
0.82
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0