



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ -
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ -
ΜΕΤΑΠΤΥΧΙΑΚΟ
ΕΦΑΡΜΟΣΜΕΝΗΣ
ΠΛΗΡΟΦΟΡΙΚΗΣ - ΕΡΓΑΣΙΑ
ΣΤΟ ΜΑΘΗΜΑ ΝΟΗΜΟΝΑ
ΣΥΣΤΗΜΑΤΑ

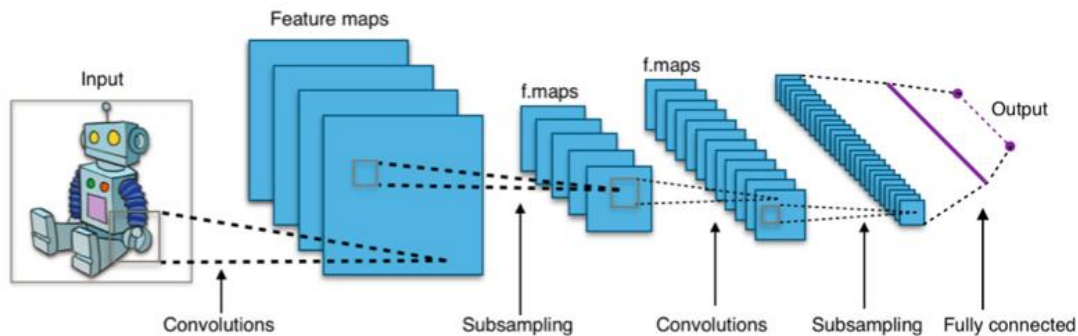
ΔΙΔΑΣΚΩΝ: ΔΡ. ΧΑΡΑΛΑΜΠΟΣ ΣΤΡΟΥΘΟΠΟΥΛΟΣ

Περιεχόμενα

Εισαγωγή στα Συνελκτικά Νευρωνικά Δίκτυα.....	2
Θεωρητικό Υπόβαθρο	2
1. Convolution - Συνέλιξη.....	3
2. Padding - Γέμισα.....	3
Activation functions.....	5
3. Pooling - Δειγματοληψία	5
4. Dropout – Ακύρωση Νευρώνων.....	6
5. Flattening - Ισοπέδωση	6
6. Fully connected layers - Πλήρες συνδεδεμένο δίκτυο.....	7
Μοντελοποίηση και υλοποίηση δικτύου αναγνώρισης ελληνικών γλυκών.	8
1. Υλοποίηση για ανάλυση εικόνων.....	8
Στατιστικά Εικόνων	9
Δείγμα γλυκών	10
Αριθμός εικόνων	11
Ιστογράμμο συχνοτήτων της μέσης τιμής των αποχρώσεων των χρωμάτων σε RGB....	11
2. Υλοποίηση για εκπαίδευση δικτύου	13
Πρότυπο προέλευσης του συνελκτικού δικτύου	13
Διαχωρισμός των εικόνων σε σύνολο εκπαίδευσης και αξιολόγησης.....	15
Οπτικοποίηση της δομή του δικτύου για ανάλυση ελληνικών γλυκών	16
Παράμετροι εκπαίδευσης του δικτύου	17
Εκπαίδευση του CNN.....	21
3. Υλοποίηση για την ανάλυση ακρίβειας και ανίχνευσης χαρακτηριστικών του CNN δικτύου	21
Εξετάζοντας την ακρίβεια του ταξινομητή.....	23
Confusion matrix - πίνακα σύγχυσης	24
Ανίχνευση χαρακτηριστικών σε κάθε επίπεδο συνέλιξης	25
Συμπεράσματα – Προτάσεις για το μέλλον.....	29

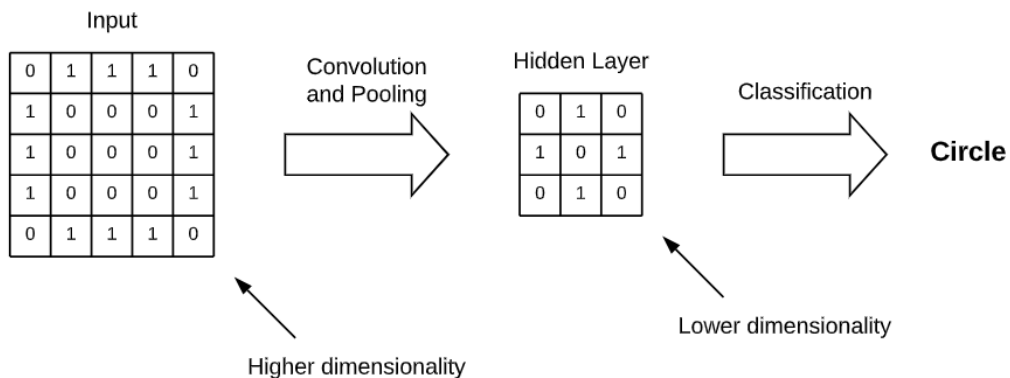
Εισαγωγή στα Συνελικτικά Νευρωνικά Δίκτυα

Τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Network CNN) είναι μια ειδική κατηγορία νευρωνικών δικτύων που έχουν δημιουργηθεί με τη δυνατότητα εξαγωγής μοναδικών χαρακτηριστικών από δεδομένα εικόνας. Για παράδειγμα, χρησιμοποιούνται στην ανίχνευση και την αναγνώριση προσώπου επειδή μπορούν να αναγνωρίσουν πολύπλοκα χαρακτηριστικά σε δεδομένα εικόνας.



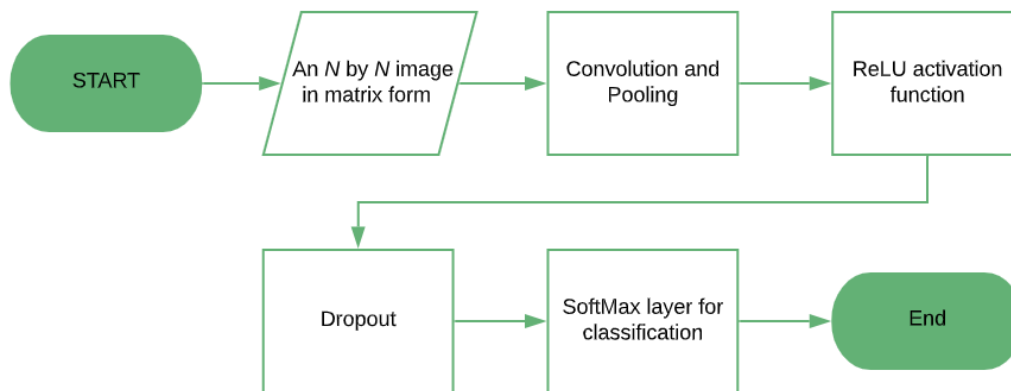
Θεωρητικό Υπόβαθρο

Τα νευρωνικά δίκτυα είναι μια υπολογιστική δομή που προσομοιάζουν τη λειτουργία των νευρώνων του εγκεφάλου και μπορούν να εκπαιδευτούν ώστε να χρησιμοποιούνται ευρέως στις μέρες σε εφαρμογές όπως αναγνώριση προτύπων σε εικόνες, πρόβλεψη κατανάλωσης ηλεκτρικής ενέργειας, για οικονομικές αναλύσεις και ανάλυση φυσικής γλώσσας. Ένα παράδειγμα για την αναγνώριση κύκλου με CNN δίκτυο.



Όπως και άλλοι τύποι νευρωνικών δικτύων, τα CNN εκπαιδεύονται από αριθμητικά δεδομένα. Επομένως, οι εικόνες που τροφοδοτούνται σε αυτά τα δίκτυα πρέπει να μετατραπούν σε μια αριθμητική αναπαράσταση. Δεδομένου ότι οι εικόνες αποτελούνται από εικονοστοιχεία, μετατρέπονται σε μια αριθμητική μορφή που μεταβιβάζεται από το CNN. Για να κατανοήσουμε πώς λειτουργεί αυτό, θα δούμε τα βήματα που εμπλέκονται στην εκπαίδευση ενός CNN.

Τα βήματα για ένα τυπικό CNN δίκτυο φαίνεται στην παρακάτω εικόνα:

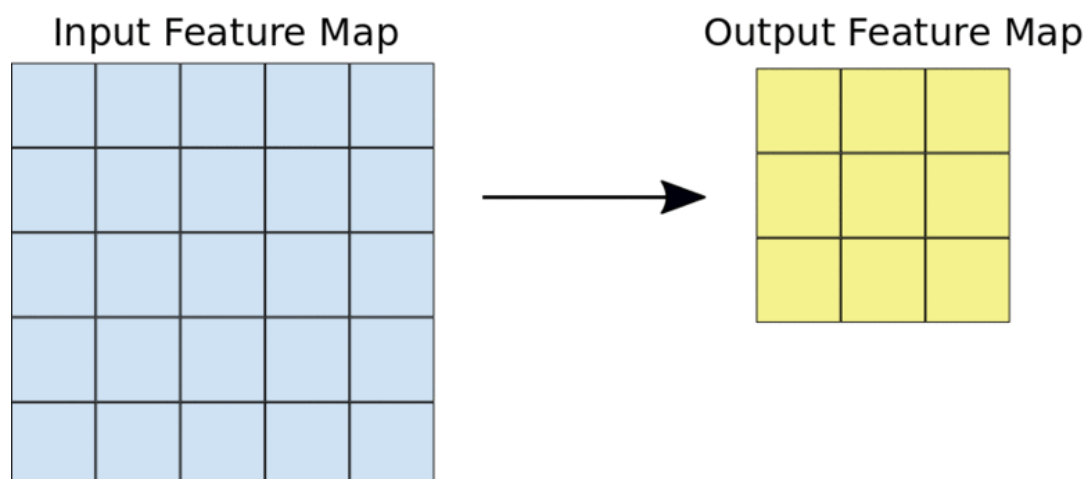


1. Convolution- Συνέλιξη

Convolution (συνέλιξη) είναι η μείωση του μεγέθους της αριθμητικής αναπαράστασης που αποστέλλεται στο CNN δίκτυο. Αυτή η διαδικασία είναι ζωτικής σημασίας, έτσι ώστε μόνο τα χαρακτηριστικά που είναι σημαντικά για την ταξινόμηση μιας εικόνας να συνεχίζουν να τροφοδοτούν το νευρωνικό δίκτυο. Εκτός από τη βελτίωση της ακρίβειας του δικτύου, αυτό διασφαλίζει επίσης ότι χρησιμοποιούνται ελάχιστοι υπολογιστικοί πόροι για την εκπαίδευση του δικτύου.

Το αποτέλεσμα της λειτουργίας της συνέλιξης περιγράφεται ως χάρτης χαρακτηριστικών, συνελιγμένο χαρακτηριστικό ή χάρτης ενεργοποίησης. Η εφαρμογή ενός ανιχνευτή χαρακτηριστικών είναι αυτό που οδηγεί σε έναν χάρτη χαρακτηριστικών. Ο ανιχνευτής χαρακτηριστικών είναι επίσης γνωστός με άλλα ονόματα όπως πυρήνας ή φίλτρο.

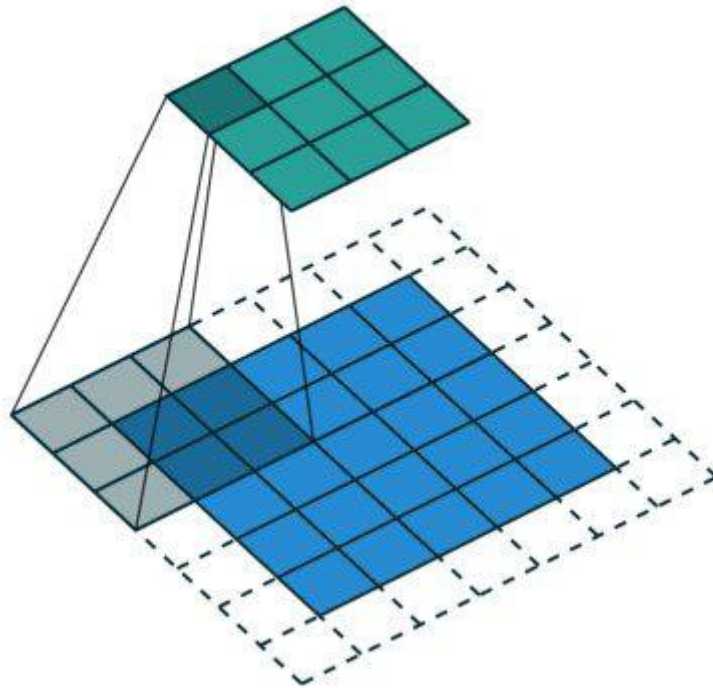
Ο πυρήνας ή φίλτρο είναι συνήθως ένας πίνακας 3 επί 3. Εκτελώντας έναν κατά στοιχείο πολλαπλασιασμό του πυρήνα με την εικόνα εισόδου και αθροίζοντας τις τιμές, βγαίνει ο χάρτης χαρακτηριστικών. Αυτό γίνεται σύροντας τον πυρήνα στην εικόνα εισόδου. Η ολίσθηση γίνεται με βήματα γνωστά ως διασκελισμοί. Τα βήματα και το μέγεθος του πυρήνα μπορούν να ρυθμιστούν χειροκίνητα κατά τη δημιουργία του CNN.



2. Padding- Γέμισα

Στις παραπάνω λειτουργίες, είδαμε ότι το μέγεθος του χάρτη χαρακτηριστικών μειώνεται ως μέρος της εφαρμογής της λειτουργίας της συνέλιξης. Τι γίνεται αν θέλετε το μέγεθος του χάρτη χαρακτηριστικών να είναι το ίδιο μέγεθος με αυτό της εικόνας εισόδου; Αυτό επιτυγχάνεται μέσω της επένδυσης.

Η συμπλήρωση (padding) περιλαμβάνει την αύξηση του μεγέθους της εικόνας εισόδου «γεμίζοντας» τις εικόνες με μηδενικά. Ως αποτέλεσμα, η εφαρμογή του φίλτρου στην εικόνα οδηγεί σε έναν χάρτη χαρακτηριστικών του ίδιου μεγέθους με την εικόνα εισόδου.



Η συμπλήρωση (padding) μειώνει τον όγκο των πληροφοριών που χάνονται στη λειτουργία συνέλιξης. Εξασφαλίζει επίσης ότι τα άκρα των εικόνων χρησιμοποιούνται συχνότερα στη λειτουργία συνέλιξης.

Στο παρακάτω παράδειγμα βλέπουμε πως γίνεται ο συνέλικτικός (convolutional) μετασχηματισμός εικόνων με 3x3 φίλτρο

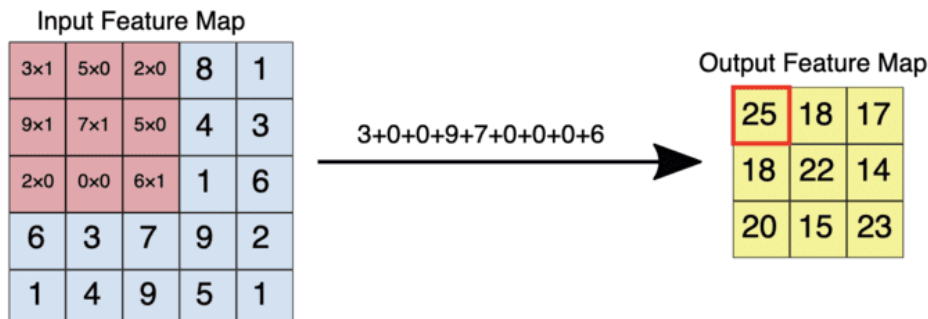
Input Feature Map

3	5	2	8	1
9	7	5	4	3
2	0	6	1	6
6	3	7	9	2
1	4	9	5	1

Convolutional Filter

1	0	0
1	1	0
0	0	1

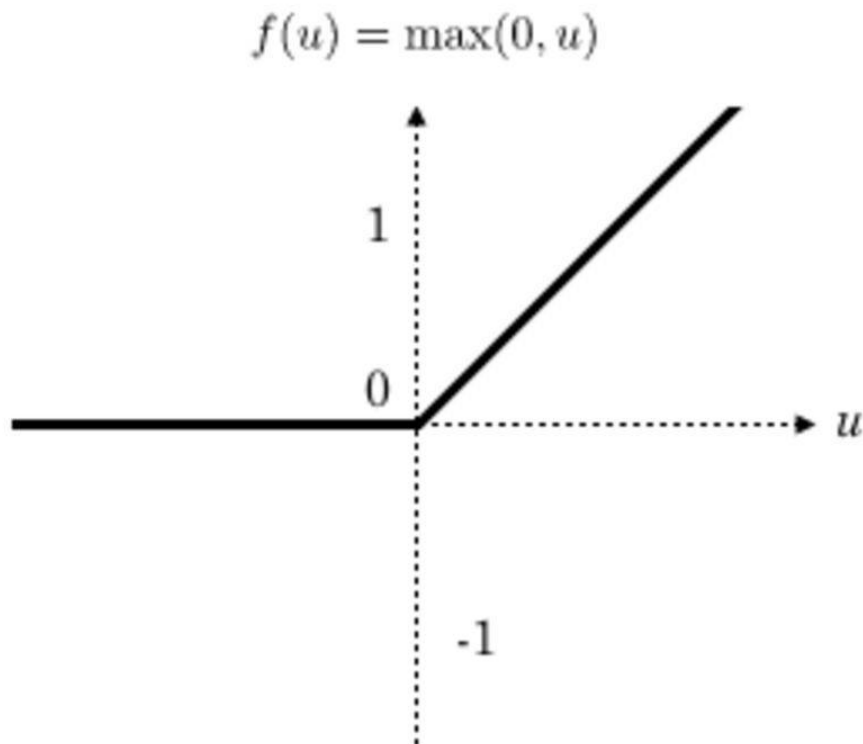
Μετά το padding, ξεκινά η διαδικασία της συνέλιξης. Το πώς θα μοιάζει ο κατά στοιχεία πολλαπλασιασμός του παραπάνω χάρτη χαρακτηριστικών και φίλτρου φαίνεται στην παρακάτω εικόνα



Activation functions

Ένας μετασχηματισμός Διορθωμένης Γραμμικής Μονάδας (Rectified Linear Unit ReLU) εφαρμόζεται μετά από κάθε λειτουργία συνέλιξης για να διασφαλιστεί η μη γραμμικότητα. Το ReLU είναι η πιο δημοφιλής λειτουργία ενεργοποίησης.

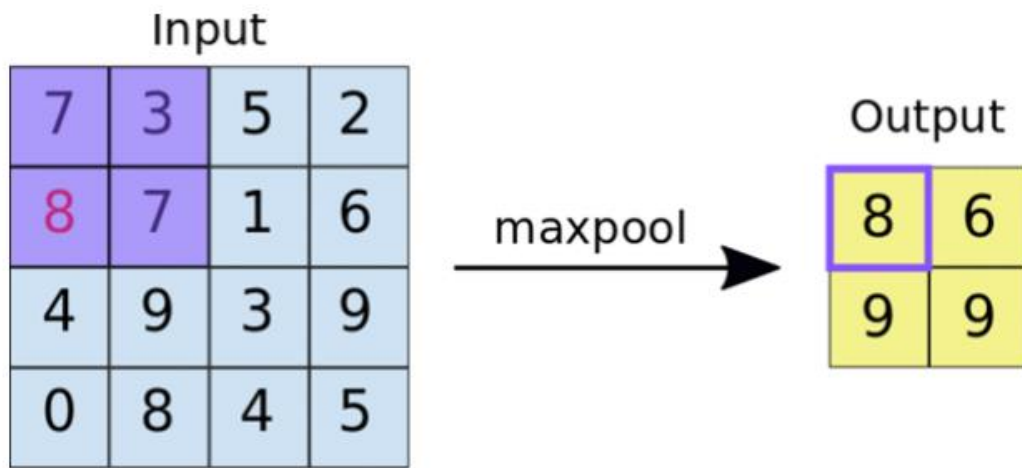
Μετά τον μετασχηματισμό, όλες οι τιμές κάτω από το μηδέν επιστρέφονται ως μηδέν ενώ οι άλλες τιμές επιστρέφονται ως έχουν.



3. Pooling- Δειγματοληψία

Σε αυτή τη λειτουργία, το μέγεθος του χάρτη χαρακτηριστικών μειώνεται περαιτέρω. Υπάρχουν διάφορες μέθοδοι συγκέντρωσης (pooling methods). Μια κοινή τεχνική είναι το max-pooling. Το μέγεθος του φίλτρου συγκέντρωσης είναι συνήθως μια μήτρα 2 επί 2. Στη max-pooling, το φίλτρο 2 επί 2 ολισθαίνει πάνω από τον χάρτη χαρακτηριστικών και επιλέγει τη μεγαλύτερη τιμή σε ένα δεδομένο πλαίσιο. Αυτή η λειτουργία οδηγεί σε έναν συγκεντρωτικό χάρτη χαρακτηριστικών.

Μια κοινή τεχνική είναι το max-pooling. Το μέγεθος του φίλτρου συγκέντρωσης είναι συνήθως μια μήτρα 2 επί 2. Στη max-pooling, το φίλτρο 2 επί 2 ολισθαίνει πάνω από τον χάρτη χαρακτηριστικών και επιλέγει τη μεγαλύτερη τιμή σε ένα δεδομένο πλαίσιο. Αυτή η λειτουργία οδηγεί σε έναν συγκεντρωτικό χάρτη χαρακτηριστικών.

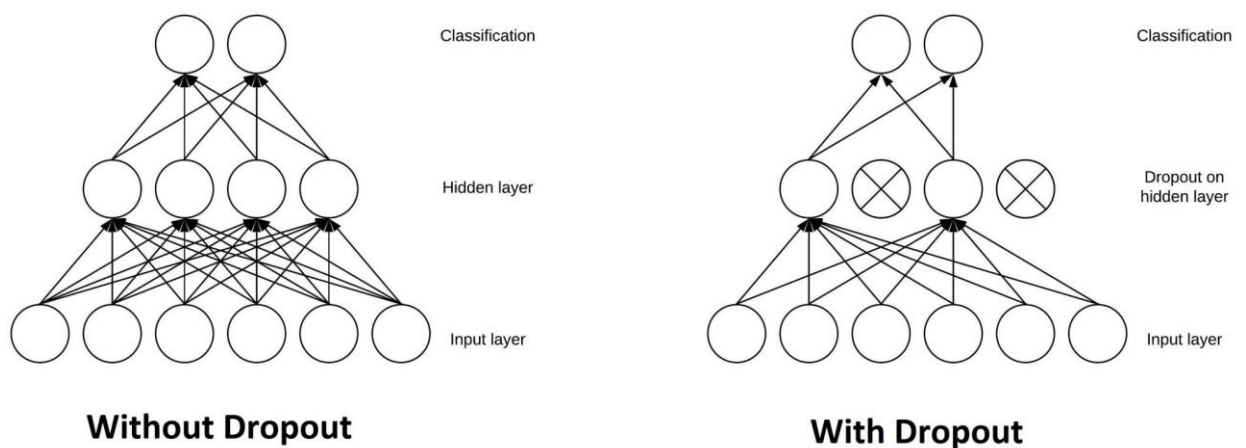


Το pooling αναγκάζει το δίκτυο να αναγνωρίσει βασικά χαρακτηριστικά στην εικόνα ανεξάρτητα από την τοποθεσία τους. Το μειωμένο μέγεθος εικόνας κάνει επίσης την εκπαίδευση του δικτύου πιο γρήγορη.

4. Dropout – Ακύρωση Νευρώνων

Ένα άλλο τυπικό χαρακτηριστικό των CNN είναι το επίπεδο Dropout. Το επίπεδο Dropout είναι μια μάσκα που ακυρώνει τη συμβολή ορισμένων νευρώνων στο επόμενο στρώμα και αφήνει αμετάβλητους όλους τους άλλους. Μπορούμε να εφαρμόσουμε ένα επίπεδο Dropout στο διάνυσμα εισόδου, οπότε ακυρώνει ορισμένα από τα χαρακτηριστικά του. αλλά μπορούμε επίσης να το εφαρμόσουμε σε ένα κρυφό στρώμα, οπότε ακυρώνει κάποιους κρυμμένους νευρώνες.

Τα επίπεδα Dropout είναι σημαντικά για την εκπαίδευση των CNN επειδή εμποδίζουν την υπερπροσαρμογή στα δεδομένα εκπαίδευσης. Εάν δεν υπάρχουν, η πρώτη παρτίδα δειγμάτων εκπαίδευσης επηρεάζει τη μάθηση με δυσανάλογα υψηλό τρόπο. Αυτό, με τη σειρά του, θα εμπόδιζε την εκμάθηση χαρακτηριστικών που εμφανίζονται μόνο σε μεταγενέστερα δείγματα ή batches:



5. Flattening- Ισοπέδωση

Η ισοπέδωση περιλαμβάνει τη μετατροπή του ομαδοποιημένου χάρτη χαρακτηριστικών σε μια ενιαία στήλη που μεταβιβάζεται στο πλήρως συνδεδεμένο επίπεδο. Αυτή είναι μια κοινή πρακτική κατά τη μετάβαση από συνελικτικά επίπεδα σε πλήρως συνδεδεμένα επίπεδα.

6. Fully connected layers - Πλήρες συνδεδεμένο δίκτυο

Στη συνέχεια, ο ισοπεδωμένος (flattened) χάρτης χαρακτηριστικών μεταβιβάζεται σε ένα πλήρως συνδεδεμένο επίπεδο. Μπορεί να υπάρχουν πολλά πλήρως συνδεδεμένα επίπεδα ανάλογα με το πρόβλημα και το δίκτυο. Το τελευταίο πλήρως συνδεδεμένο στρώμα είναι υπεύθυνο για την έξοδο της πρόβλεψης.

Μια συνάρτηση ενεργοποίησης χρησιμοποιείται στο τελικό επίπεδο ανάλογα με τον τύπο του προβλήματος. Μια σιγμοειδής ενεργοποίηση χρησιμοποιείται για δυαδική ταξινόμηση, ενώ μια συνάρτηση ενεργοποίησης softmax χρησιμοποιείται για ταξινόμηση εικόνων πολλαπλών κλάσεων.

Μοντελοποίηση και υλοποίηση δικτύου αναγνώρισης ελληνικών γλυκών.

Για την αναγνώριση γλυκών κάνοντας χρήση των συνελκτικών δικτύων θα αναπτύξουμε 3 διαφορετικές συναρτήσεις στη matlab

1. Συνάρτηση / υλοποίηση για ανάλυση εικόνων
2. Συνάρτηση / υλοποίηση για εκπαίδευση δικτύου σε matlab
3. Συνάρτηση / υλοποίηση για ανάλυση ακρίβειας του εκπαιδευμένου δικτύου στο matlab

1. Υλοποίηση για ανάλυση εικόνων

Ο παρακάτω κώδικας / υλοποίηση σε matlab είναι συνάρτηση για πρόσβαση στις εικόνες των ελληνικών γλυκών και παρουσιάζει ένα δείγμα των εικόνων, ένα ιστόγραμμα συχνотήτων με τον αριθμό των εικόνων

```
function [imds, averageHistograms] = a_exploring_sweets()

%setting the random number to be fixed
% in order to initialize the cnn's weights
rng(0);

%setting up the pathname
data = "C:\PRAXIDATA\Praxitelis\2.2 - Master\Semester
2\CNN_Sweets";

% reading the images
imds = imageDatastore(data, 'IncludeSubFolders',true,...
    'FileExtensions',[".jpg",".tif"],...
    'LabelSource','foldernames');

% useful info regarding some images
% some statistics.
numObs = length(imds.Labels);
numObsPerClass = countEachLabel(imds)
figure;
histogram(imds.Labels)
set(gca, 'TickLabelInterpreter','none')

% calculate the histogram of average RGB colors for all images
numBins = 256; % Number of bins in the histogram
accumulatedHistograms = zeros(numBins, 3); % One column for each
RGB channel
imageCount = 0;

while hasdata(imds)
    img = read(imds);
    % Split the image into RGB channels
    redChannel = img(:,:,1);
    greenChannel = img(:,:,2);
    blueChannel = img(:,:,3);

    % Calculate histograms for each channel
    redHistogram = imhist(redChannel, numBins);
    greenHistogram = imhist(greenChannel, numBins);
    blueHistogram = imhist(blueChannel, numBins);

    % Accumulate histograms
```

```

        accumulatedHistograms(:, 1) = accumulatedHistograms(:, 1) +
redHistogram;
        accumulatedHistograms(:, 2) = accumulatedHistograms(:, 2) +
greenHistogram;
        accumulatedHistograms(:, 3) = accumulatedHistograms(:, 3) +
blueHistogram;

        imageCount = imageCount + 1;
    end

    averageHistograms = accumulatedHistograms / imageCount;

    figure;
    subplot(1, 3, 1);
    bar(0:numBins-1, averageHistograms(:, 1), 'r');
    title('Red Channel');
    xlabel('Pixel Value');
    ylabel('Frequency');

    subplot(1, 3, 2);
    bar(0:numBins-1, averageHistograms(:, 2), 'g');
    title('Green Channel');
    xlabel('Pixel Value');
    ylabel('Frequency');

    subplot(1, 3, 3);
    bar(0:numBins-1, averageHistograms(:, 3), 'b');
    title('Blue Channel');
    xlabel('Pixel Value');
    ylabel('Frequency');

    sgtitle('Average Histograms of RGB Channels');
end

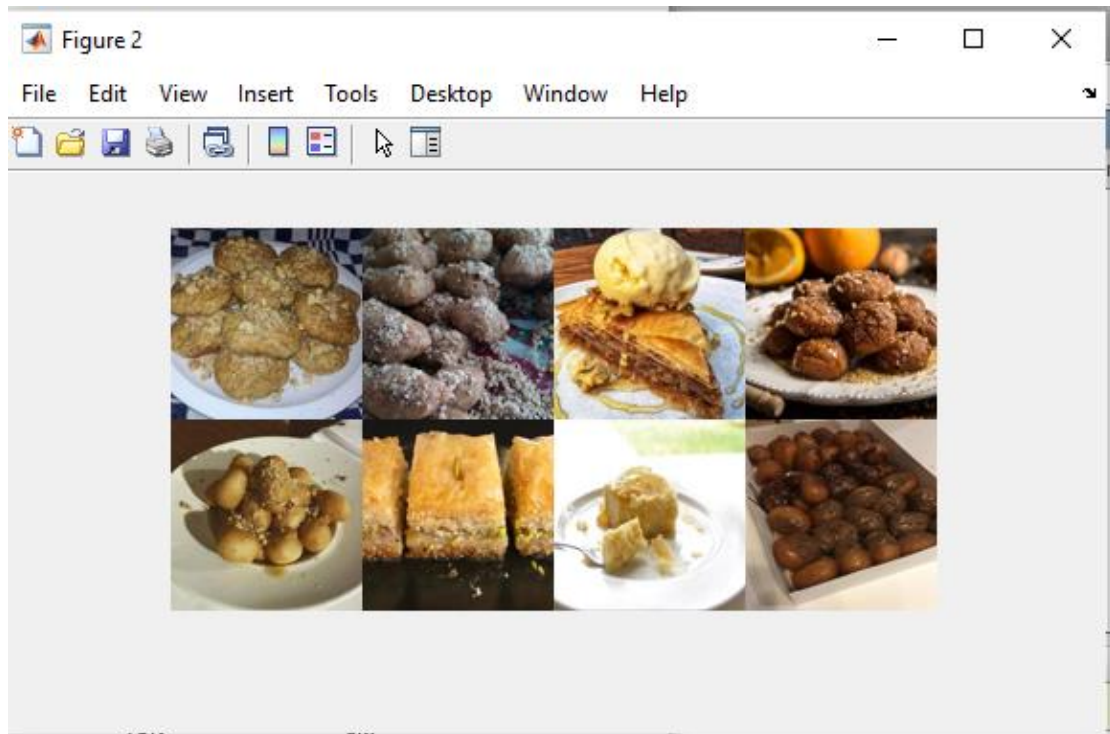
```

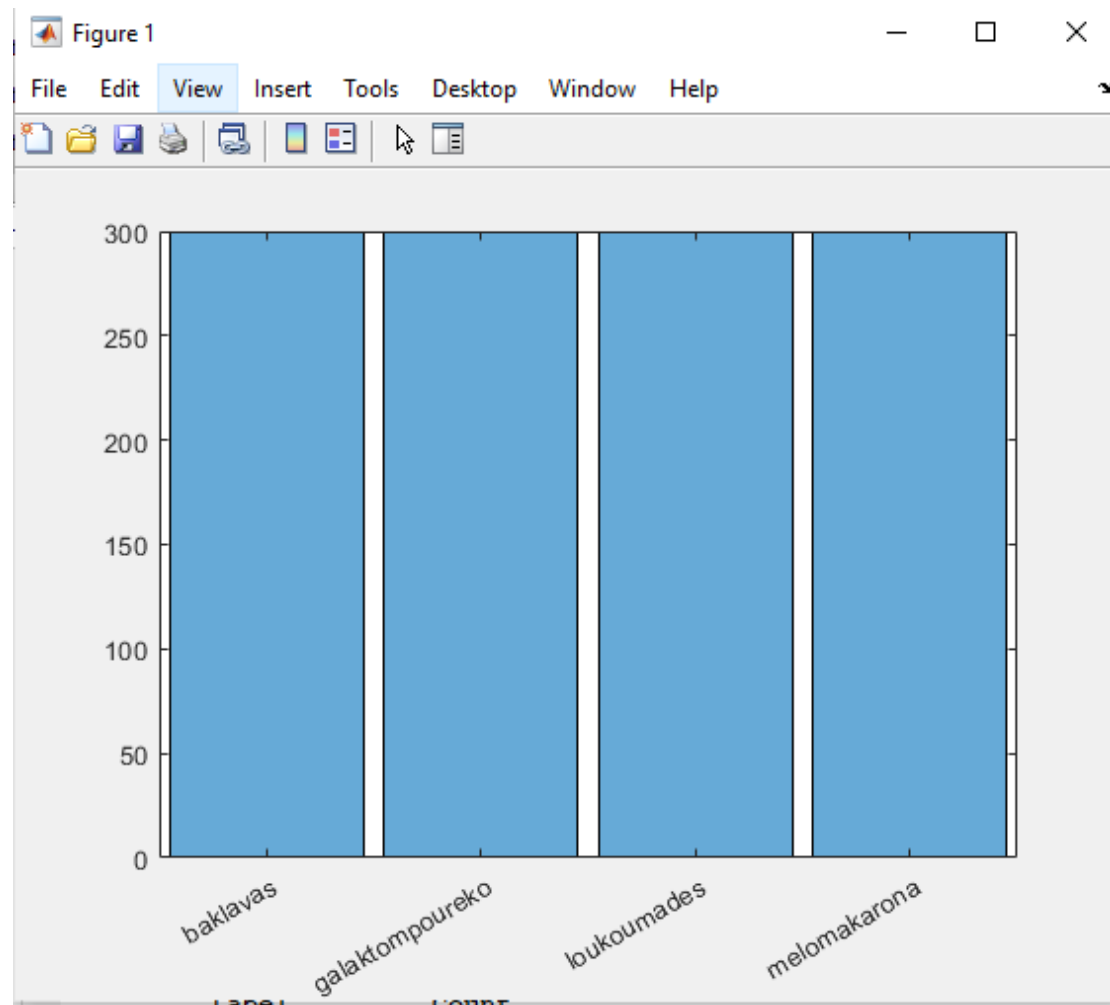
Στατιστικά Εικόνων

Έχουμε 4 κατηγορίες γλυκών, τους μπακλαβάδες, γαλακτομπούρεκα, λουκουμάδες και τα μελομακάρονα. Κάθε κατηγορία αποτελείται από 300 εικόνες, σύνολο 1200 εικόνες διαθέσιμες για εκπαίδευση.

Δείγμα γλυκών

Ας δούμε ένα δείγμα 8 διαφορετικών γλυκών από τις 1200 εικόνες.

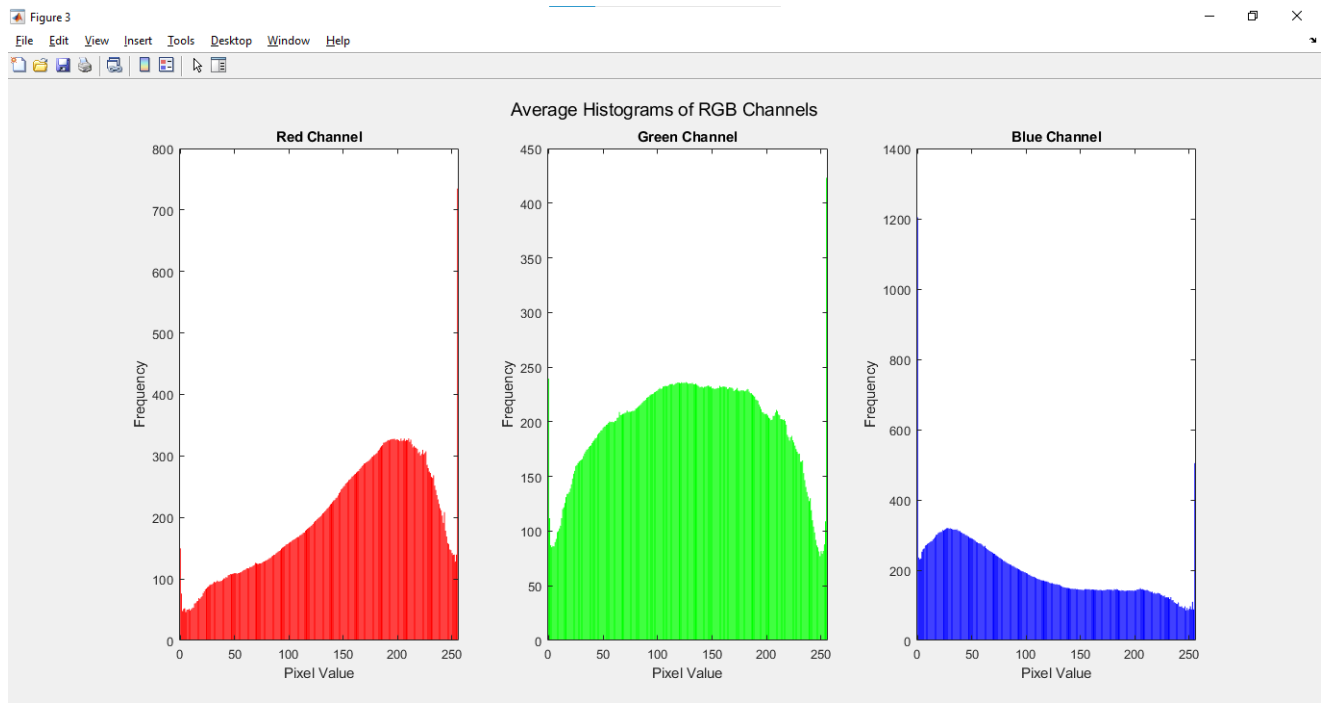




4×2 [table](#)

Label	Count
baklavas	300
galaktompourekko	300
loukoumades	300
melomakarona	300

Ιστόγραμμα συχνοτήτων της μέσης τιμής των αποχρώσεων των χρωμάτων σε RGB
 Παρατηρούμε ότι οι εικόνες μας απαρτίζονται από χρώματα που κυριαρχεί έντονα το κόκκινο. Επίσης φαίνεται ότι υπάρχουν όλες οι αποχρώσεις του πράσινου χρώματος σε όλες τις εικόνες και το μπλε χρώμα παρατηρείτε να λείπει και οι κυρίες αποχρώσεις να είναι στο ανοιχτό μπλε χρώμα.

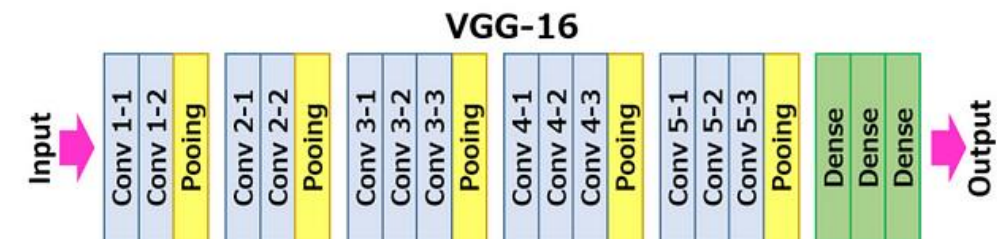


2. Υλοποίηση για εκπαίδευση δικτύου

Στη συγκεκριμένη ενότητα θα παρουσιάσουμε τον τρόπο που θα εκπαιδεύσουμε το δίκτυό μας για την ταξινόμηση ελληνικών γλυκών.

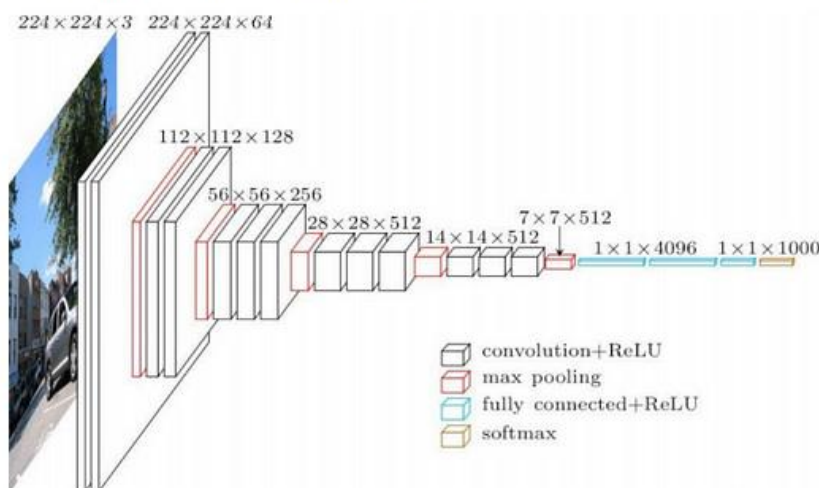
Πρότυπο προέλευσης του συνελεκτικού δικτύου

Για να δημιουργήσουμε την δομή του συνελεκτικού δικτύου μας, έγινε αναζήτηση στην επιστημονική βιβλιογραφία για αξιόπιστα δίκτυα ταξινόμησης εικόνων στα οποία η ερευνητική κοινότητα έχει εντυφλήσει. Ένα από αυτά είναι το VGG-16. Το VGG-16 είναι ένα συνελεκτικό νευρωνικό δίκτυο που έχει βάθος 16 επιπέδων. Το VGG-16 δίκτυο μπορεί να ταξινομήσει τις εικόνες σε 1000 κατηγορίες αντικειμένων, όπως πληκτρολόγιο, ποντίκι, μολύβι και πολλά ζώα σύμφωνα με τη βάση δεδομένων εικόνων [ImageNet](http://ImageNet.org).



The Architecture

The architecture depicted below is VGG16.



Στη συνέχεια θα παρουσιαστεί ο matlab κώδικας για την εκπαίδευση του CNN δικτύου.

Σε όλα τα συνελεκτικά επίπεδα θα χρησιμοποιήσουμε φίλτρο [3, 3]. Η βασική δομή του δικτύου αποτελείται από 3 βασικά στοιχεία που επαναλαμβάνονται αυξάνοντας σε κάθε επανάληψή τους τον αριθμό των φίλτρων στο επίπεδο της συνέλιξης. Αυτά τα στοιχεία είναι τα: 1) Επίπεδο Συνέλιξης, 2) batch Normalization επίπεδο και 3) Max pooling layer

1. Επίπεδο Συνέλιξης, με 16 φίλτρα των [3, 3] και γέμισμα (padding) ίσο με το μηδέν
2. Επίπεδο batch Normalization layer, που κανονικοποιεί μια μικρή παρτίδα δεδομένων σε όλες τις παρατηρήσεις για κάθε κανάλι ανεξάρτητα. Για να επιταχύνουμε την εκπαίδευση του συνελεκτικού νευρωνικού δικτύου και να μειώσουμε την ευαισθησία στην προετοιμασία του, χρησιμοποιούμε στρώματα ομαλοποίησης μεταξύ συνελεκτικών επιπέδων και μη γραμμικοτήτων, όπως τα επίπεδα ReLU.

3. Max pooling layer, Γενικά, τα επίπεδα συγκέντρωσης είναι χρήσιμα όταν θέλετε να ανιχνεύσετε ένα αντικείμενο σε μια εικόνα ανεξάρτητα από τη θέση του στην εικόνα. Η συνέπεια της προσθήκης στρώσεων συγκέντρωσης είναι η μείωση της υπερπροσαρμογής, η αυξημένη απόδοση και οι ταχύτεροι χρόνοι εκπαίδευσης σε ένα μοντέλο CNN.

Επαναλαμβάνουμε το συγκεκριμένο μοτίβο των παραπάνω τριών επιπέδων και κάθε φορά αυξάνουμε τον αριθμό των φίλτρων [3, 3] από 16 σε 32, 64, 128, 256, 512.

Στο τελευταίο επίπεδο του δικτύου αναγνώρισης γλυκών θα χρησιμοποιήσουμε 2 τελευταία επίπεδα. Το προτελευταίο επίπεδο που θα αποτελείτε από 128 νευρώνες και θα χρησιμοποιεί την ReLU συνάρτηση ενεργοποίησης και το τελευταίο επίπεδο με 4 νευρώνες

Όσες είναι και οι διαφορετικές κατηγορίες γλυκών που θέλουμε να εκπαιδύσουμε τον ταξινομητή να αναγνωρίζει και χρησιμοποιούμε στον τέλος την συνάρτηση ενεργοποίησης softmax για να μας υπολογίζει την πιθανότητα ταυτοποίησης της εικόνας του γλυκού με την κάθε κατηγορία γλυκών.

Ο matlab κώδικας για την εκπαίδευση του δικτύου είναι ο παρακάτω:

```
function [net, imdsValidation] = cnn_sweets_train(imds)

[imdsTrain, imdsValidation] = splitEachLabel(imds, 0.8);

% cnn design
batchSize = 10;
layers=[
    imageInputLayer([224 224 3]);
    convolution2dLayer([3, 3], 16, "Name", "Convolution_1",...
        'Padding', 'same');
    batchNormalizationLayer
    reluLayer;
    maxPooling2dLayer(2, 'Stride', 2);

    convolution2dLayer([3, 3], 32, "Name", "Convolution_2",...
        'Padding', 'same');
    batchNormalizationLayer
    reluLayer;
    maxPooling2dLayer([2, 2], 'Stride', 2);

    convolution2dLayer([3, 3], 64, "Name", "Convolution_3",...
        'Padding', 'same');
    batchNormalizationLayer
    reluLayer;
    maxPooling2dLayer([2, 2], 'Stride', 2);

    convolution2dLayer([3, 3], 128, "Name", "Convolution_4",...
        'Padding', 'same');
    batchNormalizationLayer
    reluLayer;
    maxPooling2dLayer([2, 2], 'Stride', 2);

    convolution2dLayer([3, 3], 256, "Name", "Convolution_5",...
        'Padding', 'same');
    batchNormalizationLayer
    reluLayer;
```

```

maxPooling2dLayer(2, 'Stride', 2);

convolution2dLayer([3, 3], 512, "Name", "Convolution_6",...
'Padding','same');
batchNormalizationLayer
reluLayer;
maxPooling2dLayer([2, 2], 'Stride', 2);

fullyConnectedLayer(128, "Name", "fullyConnected_1");
reluLayer;
fullyConnectedLayer(4, "Name", "fullyConnected_final");
softmaxLayer;
classificationLayer;];

% Cnn training options
options = trainingOptions('adam', ...
    'InitialLearnRate',0.0001,...
    'LearnRateSchedule', 'piecewise',...
    'LearnRateDropFactor', 0.1, ...
    'LearnRateDropPeriod', 3, ...
    'ExecutionEnvironment', 'gpu', ...
    'MiniBatchSize', batchSize, ...
    'MaxEpochs',20,...
    'Verbose', false,...
    'Shuffle','every-epoch', ...
    'Plots', 'training-progress',...
    'ValidationData', imdsValidation,...
    'ValidationPatience', 4, ...
    'ValidationFrequency',numel(imdsTrain.Files)/batchSize);

% training the cnn net
net = trainNetwork(imdsTrain, layers, options);
end

```

Διαχωρισμός των εικόνων σε σύνολο εκπαίδευσης και αξιολόγησης

Αρχικά θα χωρίσουμε τις 1200 εικόνες σε ομάδα εκπαίδευσης και αξιολόγησης.

Αποφασίζουμε να έχουμε 80% εικόνες για εκπαίδευση του δικτύου και τις υπόλοιπες 240 για αξιολόγηση.

Οπτικοποίηση της δομής του δικτύου για ανάλυση ελληνικών γλυκών

Στην παρακάτω εικόνα έχουμε τον πίνακα της δομής του δικτύου μας, αποτελείται από 5 στήλες τις:

1. Name, καταγράφει το όνομα του κάθε αντικειμένου του δικτύου
2. Type
3. Activations
4. Learnables
5. Total Learnables

Η εντολή για την εμφάνιση της δομής του CNN δικτύου δίνεται με την εντολή

```
% analyzing the network
analyzeNetwork(net)
```

ANALYSIS RESULT					
↑	Name	Type	Activations	Learnables	Total Learnables
1	imageinput 224x224x3 images with 'zerocenter' normalization	Image Input	224×224×3	-	0
2	Convolution_1 16 3x3x3 convolutions with stride [1 1] and padding 'same'	Convolution	224×224×16	Weights 3×3×3×16 Bias 1×1×16	448
3	batchnorm_1 Batch normalization with 16 channels	Batch Normalization	224×224×16	Offset 1×1×16 Scale 1×1×16	32
4	relu_1 ReLU	ReLU	224×224×16	-	0
5	maxpool_1 2x2 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	112×112×16	-	0
6	Convolution_2 32 3x3x16 convolutions with stride [1 1] and padding 'same'	Convolution	112×112×32	Weights 3×3×16×32 Bias 1×1×32	4640
7	batchnorm_2 Batch normalization with 32 channels	Batch Normalization	112×112×32	Offset 1×1×32 Scale 1×1×32	64
8	relu_2 ReLU	ReLU	112×112×32	-	0
9	maxpool_2 2x2 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	56×56×32	-	0
10	Convolution_3 64 3x3x32 convolutions with stride [1 1] and padding 'same'	Convolution	56×56×64	Weights 3×3×32×64 Bias 1×1×64	18496
11	batchnorm_3 Batch normalization with 64 channels	Batch Normalization	56×56×64	Offset 1×1×64 Scale 1×1×64	128
12	relu_3 ReLU	ReLU	56×56×64	-	0
13	maxpool_3 2x2 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	28×28×64	-	0
14	Convolution_4 128 3x3x64 convolutions with stride [1 1] and padding 'same'	Convolution	28×28×128	Weights 3×3×64×128 Bias 1×1×128	73856
15	batchnorm_4 Batch normalization with 128 channels	Batch Normalization	28×28×128	Offset 1×1×128 Scale 1×1×128	256
16	relu_4 ReLU	ReLU	28×28×128	-	0
17	maxpool_4 2x2 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	14×14×128	-	0
18	Convolution_5 256 3x3x128 convolutions with stride [1 1] and padding 'same'	Convolution	14×14×256	Weights 3×3×128×256 Bias 1×1×256	295168
19	batchnorm_5 Batch normalization with 256 channels	Batch Normalization	14×14×256	Offset 1×1×256 Scale 1×1×256	512
20	relu_5 ReLU	ReLU	14×14×256	-	0
21	maxpool_5 2x2 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	7×7×256	-	0
22	Convolution_6 512 3x3x256 convolutions with stride [1 1] and padding 'same'	Convolution	7×7×512	Weights 3×3×256×512 Bias 1×1×512	1180160
23	batchnorm_6 Batch normalization with 512 channels	Batch Normalization	7×7×512	Offset 1×1×512 Scale 1×1×512	1024
24	relu_6 ReLU	ReLU	7×7×512	-	0
25	maxpool_6 2x2 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	3×3×512	-	0
26	fullyConnected_1 128 fully connected layer	Fully Connected	1×1×128	Weights 128×4608 Bias 128×1	589952
27	relu_7 ReLU	ReLU	1×1×128	-	0
28	fullyConnected_final 4 fully connected layer	Fully Connected	1×1×4	Weights 4×128 Bias 4×1	516

Εικόνα 1: Δομή του δικτύου ταξινόμησης γλυκών

Παράμετροι εκπαίδευσης του δικτύου

Optimizer: adam

Ο Adam (Adaptive Moment Estimation) είναι ένας αλγόριθμος βελτιστοποίησης που χρησιμοποιείται στο πλαίσιο της μηχανικής μάθησης και της εκπαίδευσης νευρωνικών δικτύων. Ο Adam συνδυάζει στοιχεία από δύο άλλους αλγορίθμους βελτιστοποίησης, τον AdaGrad και τον RMSprop.

Ο Adam χρησιμοποιείται για την ενημέρωση των βαρών ενός μοντέλου κατά την εκπαίδευση. Με βάση την κίνηση (momentum) και την προσαρμοστική εκτίμηση της διόρθωσης διέγερσης (adaptive correction), ο Adam είναι σε θέση να ενημερώνει τα βάρη του μοντέλου με βάση την παρούσα κλίση (gradient) και την ιστορία των κλίσεων.

Ο Adam είναι γνωστός για την αποδοτικότητά του στην εκπαίδευση νευρωνικών δικτύων, καθώς συνδυάζει την ανεξαρτησία από τον ρυθμό μάθησης, την προσαρμοστικότητα στον κάθε παράμετρο, και την αποτελεσματικότητα στην εξερεύνηση του χώρου των παραμέτρων.

Συνοπτικά, ο Adam optimizer είναι ένας αλγόριθμος βελτιστοποίησης που χρησιμοποιείται για την εκπαίδευση νευρωνικών δικτύων, συνδυάζοντας στοιχεία από τους αλγορίθμους AdaGrad και RMSprop για να προσαρμόσει αποδοτικά τα βάρη του μοντέλου κατά την εκπαίδευση.

Learning rate: 0.0001

Ο ρυθμός μάθησης (learning rate) είναι ένα χρηματικό ποσοστό που καθορίζει το βήμα ή το μέγεθος της αλλαγής που γίνεται στις παραμέτρους ενός μοντέλου κατά τη διαδικασία εκπαίδευσης. Ο ρυθμός μάθησης καθορίζει πόσο γρήγορα θα συγκλίνουν οι παράμετροι του μοντέλου προς τη βέλτιστη λύση κατά τη διάρκεια της εκπαίδευσης.

Αν ο ρυθμός μάθησης είναι υψηλός, η αλλαγή στις παραμέτρους θα είναι μεγάλη σε κάθε επανάληψη της διαδικασίας εκπαίδευσης. Αυτό μπορεί να οδηγήσει σε ασταθή σύγκλιση ή αναπόδραστη από το αποδεκτό εύρος της βέλτιστης λύσης. Από την άλλη πλευρά, αν ο ρυθμός μάθησης είναι χαμηλός, η εκπαίδευση μπορεί να είναι πολύ αργή και να απαιτεί περισσότερο χρόνο για να φτάσει σε μια καλή λύση.

Η επιλογή του κατάλληλου ρυθμού μάθησης είναι σημαντική και πρέπει να γίνει πειραματικά, δοκιμάζοντας διάφορες τιμές και παρακολουθώντας την απόδοση του μοντέλου σε ένα σύνολο επικύρωσης. Συνήθως, αρχίζει με έναν σχετικά μεγάλο ρυθμό μάθησης για γρήγορη σύγκλιση και στη συνέχεια μειώνεται καθώς η εκπαίδευση προχωρά, προκειμένου να επιτευχθεί μεγαλύτερη ακρίβεια.

Learning Rate schedule: piecewise

Το Learning Rate schedule (πρόγραμμα ρυθμού μάθησης) αναφέρεται στον τρόπο με τον οποίο αλλάζει ο ρυθμός μάθησης κατά τη διάρκεια της εκπαίδευσης ενός μοντέλου μηχανικής μάθησης. Αντί να χρησιμοποιείται ένας σταθερός ρυθμός μάθησης καθ' όλη τη διαδικασία εκπαίδευσης, το Learning Rate schedule επιτρέπει την αλλαγή του ρυθμού μάθησης κατάλληλα κατά τη διάρκεια της εκπαίδευσης.

Οι πιο συνηθισμένες μέθοδοι για το Learning Rate schedule περιλαμβάνουν τα παρακάτω:

1. Σταθερός ρυθμός μάθησης: Ο ρυθμός μάθησης παραμένει σταθερός κατά τη διάρκεια της εκπαίδευσης. Αυτή είναι η πιο απλή μέθοδος, αλλά μπορεί να είναι δύσκολο να επιλεγεί ο βέλτιστος σταθερός ρυθμός μάθησης για όλη τη διαδικασία.

2. Προκαθορισμένο πρόγραμμα: Ορίζονται προκαθορισμένοι ρυθμοί μάθησης για συγκεκριμένα επαναληπτικά βήματα ή εποχές. Για παράδειγμα, μπορεί να οριστεί ένα πρόγραμμα όπου ο ρυθμός μάθησης μειώνεται κατά το 50% ανά 10 επαναλήψεις.

3. Προσαρμοστικός ρυθμός μάθησης: Ο ρυθμός μάθησης προσαρμόζεται δυναμικά κατά τη διάρκεια της εκπαίδευσης, βάσει κάποιου κανόνα ή αλγορίθμου. Γνωστοί αλγόριθμοι περιλαμβάνουν τον αλγόριθμο Adam που αναφέρθηκε προηγουμένως.

Το σωστό Learning Rate schedule μπορεί να βοηθήσει στη βελτίωση της απόδοσης του μοντέλου, επιτρέποντας τη σύγκλιση σε μια βέλτιστη λύση με λιγότερες επαναλήψεις ή εποχές εκπαίδευσης.

LearnRateDropFactor: 10% ή 0.1

Το LearnRateDropFactor είναι ένας παράγοντας που χρησιμοποιείται στο πλαίσιο του Learning Rate schedule για τη μείωση του ρυθμού μάθησης κατά έναν συγκεκριμένο παράγοντα. Συγκεκριμένα, όταν χρησιμοποιείται η τεχνική του Learning Rate schedule, ο LearnRateDropFactor καθορίζει το ποσοστό μείωσης του ρυθμού μάθησης.

Κατά τη διάρκεια της εκπαίδευσης, όταν φτάσει ένα ορισμένο σημείο ή επανάληψη στο Learning Rate schedule, ο ρυθμός μάθησης μειώνεται κατά τον παράγοντα LearnRateDropFactor. Αυτό μπορεί να επιτευχθεί με την απλή πολλαπλασιαστική μείωση του ρυθμού μάθησης κατά έναν παράγοντα LearnRateDropFactor, όπως για παράδειγμα:

Νέος ρυθμός μάθησης = Παλιός ρυθμός μάθησης * LearnRateDropFactor

Ο LearnRateDropFactor καθορίζεται από τον χρήστη και επιλέγεται ανάλογα με την εφαρμογή και τις απαιτήσεις του μοντέλου. Ένας κοινός παράγοντας LearnRateDropFactor είναι 0,1, που σημαίνει ότι ο ρυθμός μάθησης μειώνεται κατά 90% σε κάθε επανάληψη ή στο καθορισμένο σημείο. Ωστόσο, ο παράγοντας μπορεί να είναι διαφορετικός ανάλογα με τις ανάγκες του προβλήματος.

LearnRateDropPeriod

Η παράμετρος "LearnRateDropPeriod" στο MATLAB Deep Learning αναφέρεται στον αριθμό των εποχών εκπαίδευσης μεταξύ των μειώσεων του ρυθμού μάθησης (learning rate) κατά τη χρήση της παραμέτρου "LearnRateDropFactor".

Κατά την εκπαίδευση ενός μοντέλου μηχανικής μάθησης, είναι συνήθης πρακτική να μειώνεται ο ρυθμός μάθησης κατά έναν παράγοντα για να βελτιωθεί η σύγκλιση και η απόδοση του μοντέλου. Η παράμετρος "LearnRateDropPeriod" καθορίζει τον αριθμό των εποχών εκπαίδευσης πριν από κάθε μείωση του ρυθμού μάθησης.

Για παράδειγμα, εάν η τιμή της παραμέτρου "LearnRateDropPeriod" είναι 10 και η παράμετρος "LearnRateDropFactor" είναι 0.1, τότε ο ρυθμός μάθησης θα μειωθεί κατά έναν παράγοντα 0.1 κάθε 10 εποχές εκπαίδευσης.

Η χρήση των παραμέτρων "LearnRateDropPeriod" και "LearnRateDropFactor" επιτρέπει την προσαρμογή του ρυθμού μάθησης κατά τη διάρκεια της εκπαίδευσης για βελτιωμένη απόδοση και σύγκλιση του μοντέλου.

Execution environment: GPU

Το περιβάλλον εκτέλεσης (execution environment) αναφέρεται στο σύνολο των συνθηκών και των πόρων που είναι διαθέσιμοι για την εκτέλεση ενός προγράμματος. Αυτό περιλαμβάνει το λογισμικό, το υλικό, το λειτουργικό σύστημα και τις βιβλιοθήκες που είναι αναγκαίες για την εκτέλεση του προγράμματος.

Συγκεκριμένα για το MATLAB, το περιβάλλον εκτέλεσης περιλαμβάνει το MATLAB Runtime, το οποίο είναι απαραίτητο για να εκτελέσετε MATLAB κώδικα εκτός από το MATLAB περιβάλλον επεξεργασίας. Το MATLAB Runtime παρέχει τις απαραίτητες βιβλιοθήκες και τους πόρους που απαιτούνται για την εκτέλεση του κώδικα MATLAB σε ένα περιβάλλον όπως ένα αυτόνομο εκτελέσιμο αρχείο ή ένα περιβάλλον εκτέλεσης MATLAB που είναι εγκατεστημένο σε μια συγκεκριμένη πλατφόρμα.

Για την εκπαίδευση του συνελκτικού δικτύου θα χρησιμοποιήσουμε τους πόρους της κάρτας γραφικών.

MiniBatchSize

Το MiniBatchSize αναφέρεται στον αριθμό των δειγμάτων δεδομένων που χρησιμοποιούνται για την εκπαίδευση ενός μοντέλου μηχανικής μάθησης σε κάθε επανάληψη της διαδικασίας εκπαίδευσης. Κατά την εκπαίδευση ενός μοντέλου, τα δεδομένα συχνά χωρίζονται σε μικρότερα υποσύνολα που ονομάζονται mini-batches. Το MiniBatchSize καθορίζει τον αριθμό των δειγμάτων που περιλαμβάνονται σε κάθε mini-batch.

Η χρήση του MiniBatchSize έχει πολλά πλεονεκτήματα:

1. Επιτάχυνση της εκπαίδευσης: Αντί να εκπαιδεύετε το μοντέλο με όλα τα δεδομένα εκπαίδευσης ταυτόχρονα, η εκπαίδευση με mini-batches επιτρέπει την παράλληλη επεξεργασία πολλαπλών mini-batches, βελτιώνοντας την απόδοση.
2. Αποθήκευση μνήμης: Εφόσον δεν χρειάζεται να φορτώνονται όλα τα δεδομένα εκπαίδευσης ταυτόχρονα, η χρήση mini-batches μειώνει την απαίτηση μνήμης.
3. Εξομάλυνση της εκπαίδευσης: Η εκπαίδευση με mini-batches μπορεί να βοηθήσει στην εξομάλυνση της διαδικασίας εκπαίδευσης, καθώς τα δεδομένα ενός mini-batch είναι μια προσέγγιση του γενικού συνόλου δεδομένων εκπαίδευσης.

MaxEpochs

Η παράμετρος MaxEpochs (ή μέγιστος αριθμός εποχών) αναφέρεται στον αριθμό των επαναλήψεων που θα γίνουν κατά την εκπαίδευση ενός μοντέλου μηχανικής μάθησης. Μία εποχή αντιστοιχεί σε μια πλήρη προβολή και επεξεργασία του συνόλου δεδομένων εκπαίδευσης.

Ορίζοντας την παράμετρο MaxEpochs, καθορίζετε τον αριθμό των εποχών που το μοντέλο σας θα εκπαιδευτεί. Κατά την εκπαίδευση, το μοντέλο ενημερώνεται σταδιακά, προσαρμόζοντας τα βάρη και τις παραμέτρους του, προκειμένου να μειώσει το σφάλμα και να βελτιώσει την απόδοση στα δεδομένα εκπαίδευσης.

Ο μέγιστος αριθμός εποχών (MaxEpochs) ορίζει πόσες φορές το μοντέλο θα εκπαιδευτεί στο σύνολο των δεδομένων εκπαίδευσης. Μετά τον ορισμένο αριθμό των εποχών, η εκπαίδευση συνήθως σταματά. Η επιλογή ενός κατάλληλου αριθμού εποχών εξαρτάται από το πρόβλημα, τον τύπο του μοντέλου και την εμπειρία του εκπαιδευτή. Πολλές φορές, ο αριθμός των εποχών είναι ένας υπέρ-παράμετρος που πρέπει να προσαρμόζεται πειραματικά για να επιτευχθεί καλή επίδοση του μοντέλου.

Shuffle

Στο MATLAB Deep Learning, η παράμετρος "Shuffle" αναφέρεται στην ανακάτεμα των δεδομένων εκπαίδευσης πριν από κάθε εποχή εκπαίδευσης. Όταν η παράμετρος "Shuffle" είναι ενεργοποιημένη, τα δεδομένα εκπαίδευσης ανακατεύονται πριν την έναρξη μιας νέας εποχής, έτσι ώστε τα δείγματα να παρουσιάζονται στο μοντέλο σε μια τυχαία σειρά κατά τη διάρκεια της εκπαίδευσης.

Το ανακάτεμα των δεδομένων εκπαίδευσης είναι σημαντική για να αποφευχθεί η πιθανότητα οι δείγματα να παρουσιαστούν στο μοντέλο με κάποια συγκεκριμένη σειρά που μπορεί να επηρεάσει την εκπαίδευση και την απόδοση του μοντέλου. Η ανακάτεμα των δεδομένων επιτρέπει στο μοντέλο να εκπαιδευτεί με μια πιο ανεξάρτητη και αντιπροσωπευτική αναπαράσταση του συνόλου δεδομένων εκπαίδευσης.

ValidationData

Η παράμετρος "ValidationData" στο MATLAB Deep Learning αναφέρεται στα δεδομένα επικύρωσης που χρησιμοποιούνται για την αξιολόγηση της απόδοσης του μοντέλου κατά τη διάρκεια της εκπαίδευσης. Τα δεδομένα επικύρωσης είναι ένα ανεξάρτητο σύνολο δεδομένων που δεν χρησιμοποιούνται για την εκπαίδευση του μοντέλου, αλλά χρησιμοποιούνται για να αξιολογήσουν την απόδοσή του σε ένα στάδιο επικύρωσης.

Κατά την εκπαίδευση ενός μοντέλου, είναι συνηθισμένο να χρησιμοποιούνται τρία σύνολα δεδομένων: το σύνολο δεδομένων εκπαίδευσης, το σύνολο δεδομένων επικύρωσης και το σύνολο δεδομένων ελέγχου (ή δοκιμής). Το σύνολο δεδομένων επικύρωσης χρησιμοποιείται για να εκτιμήσει την απόδοση του μοντέλου κατά τη διάρκεια της εκπαίδευσης και για να βοηθήσει στην επιλογή των βέλτιστων υπερπαραμέτρων.

ValidationPatience

Η παράμετρος "ValidationPatience" στο MATLAB Deep Learning αναφέρεται στην παύση της εκπαίδευσης εάν η απόδοση του μοντέλου στα δεδομένα επικύρωσης δεν βελτιώνεται για έναν προκαθορισμένο αριθμό εποχών συνεχόμενα.

Κατά τη διάρκεια της εκπαίδευσης, η παράμετρος "ValidationPatience" ελέγχει την απόδοση του μοντέλου στα δεδομένα επικύρωσης και συγκρίνει την απόδοση της τρέχουσας εποχής με την απόδοση των προηγούμενων εποχών. Εάν η απόδοση στα δεδομένα επικύρωσης δεν βελτιώνεται για έναν προκαθορισμένο αριθμό εποχών συνεχόμενα, η εκπαίδευση διακόπτεται, προκειμένου να αποφευχθεί η υπερεκπαίδευση (overfitting).

Η παράμετρος "ValidationPatience" συνήθως συνδυάζεται με την παράμετρο "EarlyStopping" για τον έλεγχο της πρόωρης διακοπής της εκπαίδευσης. Εάν η απόδοση στα δεδομένα επικύρωσης δεν βελτιώνεται για έναν προκαθορισμένο αριθμό εποχών συνεχόμενα, η εκπαίδευση διακόπτεται πρόωρα.

Για να ορίσετε την παράμετρο "ValidationPatience" στο MATLAB Deep Learning, μπορείτε να χρησιμοποιήσετε την αντίστοιχη παράμετρο κατά την κατασκευή του αντικειμένου του εκπαιδευτή (trainer).

ValidationFrequency

Η παράμετρος "ValidationFrequency" στο MATLAB Deep Learning αναφέρεται στον αριθμό των επαναλήψεων εκπαίδευσης πριν γίνει εκτελεστεί η αξιολόγηση του μοντέλου στα δεδομένα επικύρωσης.

Κατά τη διάρκεια της εκπαίδευσης, η παράμετρος "ValidationFrequency" ελέγχει τον αριθμό των επαναλήψεων εκπαίδευσης (ή ενημερώσεων των βαρών) που πρέπει να εκτελεστούν πριν γίνει η αξιολόγηση του μοντέλου στα δεδομένα επικύρωσης. Αυτό επιτρέπει την προσαρμογή της συχνότητας αξιολόγησης του μοντέλου κατά τη διάρκεια της εκπαίδευσης.

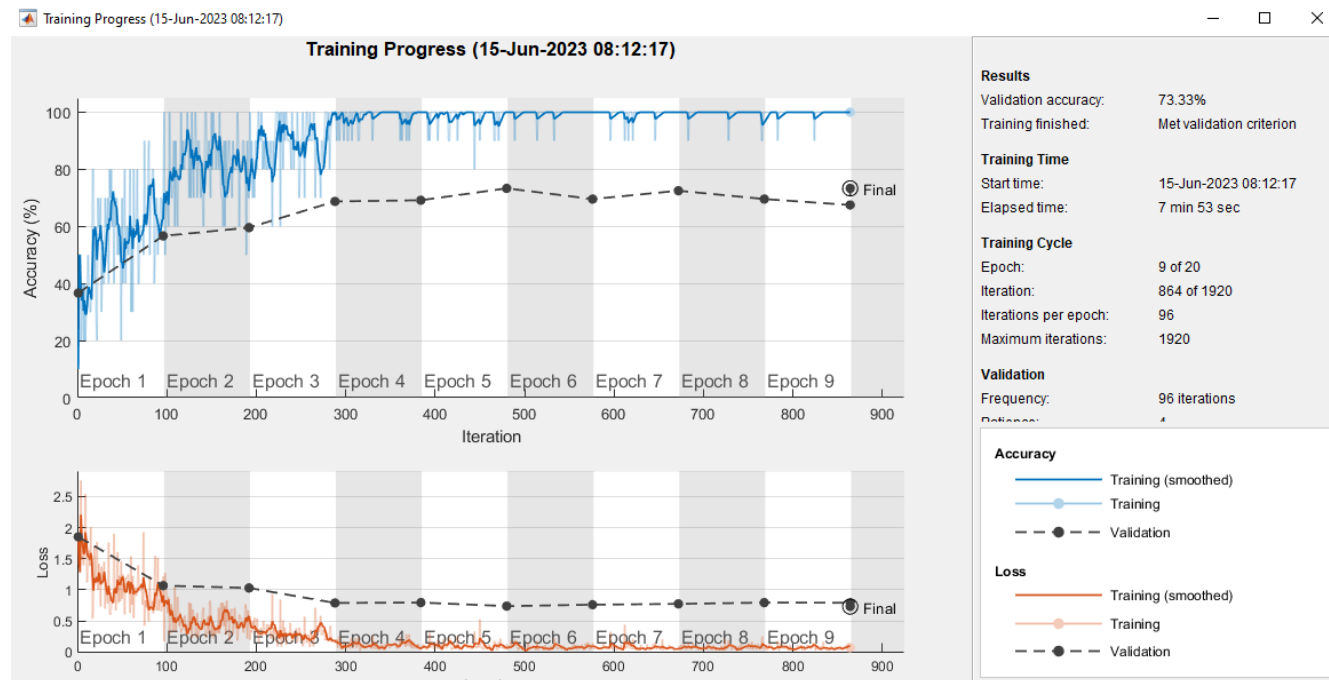
Για παράδειγμα, εάν η τιμή της παραμέτρου "ValidationFrequency" είναι 10, τότε η αξιολόγηση του μοντέλου στα δεδομένα επικύρωσης θα γίνει κάθε 10 επαναλήψεις εκπαίδευσης.

Η χρήση της παραμέτρου "ValidationFrequency" επιτρέπει την ελέγχου της απόδοσης του μοντέλου σε τακτά διαστήματα, μειώνοντας τον υπολογιστικό φόρτο και εξοικονομώντας χρόνο κατά τη διάρκεια της εκπαίδευσης.

Για να ορίσετε την παράμετρο "ValidationFrequency" στο MATLAB Deep Learning, μπορείτε να χρησιμοποιήσετε την αντίστοιχη παράμετρο κατά την κατασκευή του αντικειμένου του εκπαιδευτή (trainer).

Εκπαίδευση του CNN

Η εκπαίδευση του δικτύου μας διαρκεί για 9 εποχές εκπαίδευσης, δηλαδή εκπαιδεύεται στα δεδομένα εκπαίδευσης έως 9 φορές και σε κάθε εποχή παρατηρούμε ότι η απόδοση της ακρίβειας του δικτύου στο σύνολο εκπαίδευσης να βελτιώνεται φτάνοντας το 100%, ωστόσο η ακρίβεια στο σύνολο αξιολόγησης να σταθεροποιείτε κοντά στο 73%



3. Υλοποίηση για την ανάλυσης ακρίβειας και ανίχνευσης χαρακτηριστικών του CNN δικτύου

Στο τελευταίο μέρος της εργασίας θα αξιολογήσουμε τα αποτελέσματα της ακρίβειας του ταξινομητή που δημιουργήθηκε προηγουμένως και την εξαγωγή χαρακτηριστικών των εικόνων που δημιουργούνται από τα συνελκτικά επίπεδα.

Ο matlab κώδικας για για την ανάλυσης ακρίβειας και ανίχνευσης χαρακτηριστικών του CNN δικτύου είναι ο εξής:

```
function cnn_sweets_inspection(net, imdsValidation)

% inspecting the cnn net upon the validation set
Prdigits = classify(net, imdsValidation);
digits = imdsValidation.Labels;
accuracy = 100 * sum(Prdigits==digits)/numel(digits);
disp("accuracy after training on validation set: " + accuracy
+"%")

%plotting the confussion matrix
plotconfusion(imdsValidation.Labels, Prdigits)

% analyzing the network
analyzeNetwork(net)
```



```

%Visualize Features of a Convolutional Neural Network
% Convolution layer 1
layer = 2;
name = net.Layers(layer).Name;
channels = 1:16;
I = deepDreamImage(net,name,channels, ...
    'PyramidLevels',1);

figure
I = imtile(I,'ThumbnailSize',[64 64]);
imshow(I)
title(['Layer ',name,' Features'],'Interpreter','none')

% Convolution layer 2
layer = 6;
name = net.Layers(layer).Name;
channels = 1:32;
I = deepDreamImage(net,name,channels, ...
    'PyramidLevels',1);

figure
I = imtile(I,'ThumbnailSize',[64 64]);
imshow(I)
title(['Layer ',name,' Features'],'Interpreter','none')

% Convolution layer 3
layer = 10;
name = net.Layers(layer).Name;
channels = 1:64;
I = deepDreamImage(net,name,channels, ...
    'PyramidLevels',1);

figure
I = imtile(I,'ThumbnailSize',[64 64]);
imshow(I)
title(['Layer ',name,' Features'],'Interpreter','none')

% Convolution layer 4
layer = 14;
name = net.Layers(layer).Name;
channels = 1:128;
I = deepDreamImage(net,name,channels, ...
    'PyramidLevels',1);

figure
I = imtile(I,'ThumbnailSize',[64 64]);
imshow(I)
title(['Layer ',name,' Features'],'Interpreter','none')

% Convolution layer 5
layer = 18;
name = net.Layers(layer).Name;
channels = 1:256;
I = deepDreamImage(net,name,channels, ...
    'PyramidLevels',1);

figure
I = imtile(I,'ThumbnailSize',[64 64]);
imshow(I)
title(['Layer ',name,' Features'],'Interpreter','none')

% Convolution layer 6
% cannot visualize features due to current GPU's

```

```
% memory limitations (upto 2GB)

% last full connected layer
layer = 28;
name = net.Layers(layer).Name;
channels = 1:4;
I = deepDreamImage(net,name,channels, ...
    'Verbose',true, ...
    'NumIterations',100, ...
    'PyramidLevels',2);

figure
I = imtile(I,'ThumbnailSize',[64 64]);
imshow(I)
name = net.Layers(layer).Name;
title(['Layer ',name,' Features'])

end
```

Εξετάζοντας την ακρίβεια του ταξινομητή

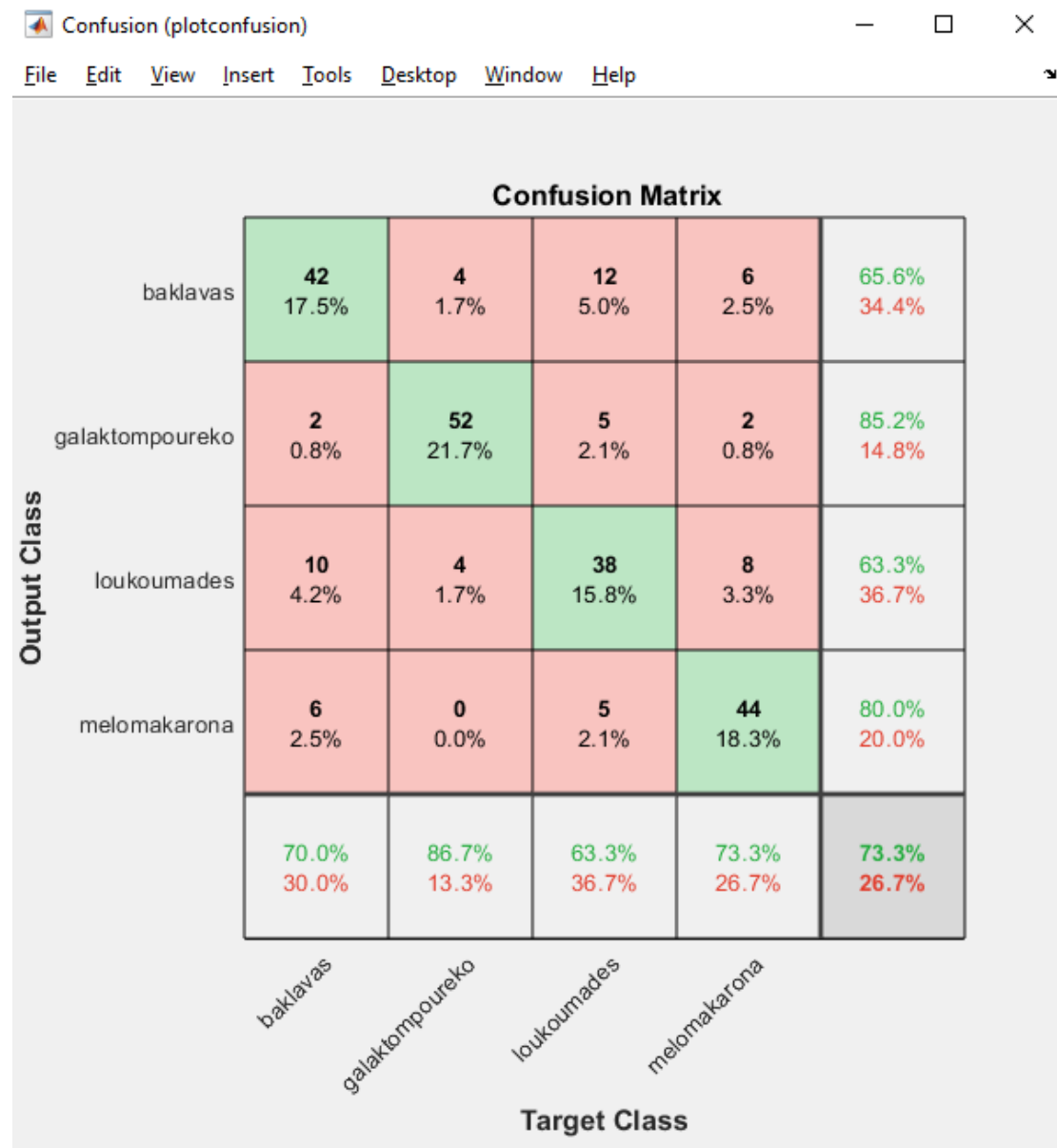
Η ακρίβεια του ταξινομητή μας στο σύνολο εικόνων που κρατήθηκαν προς αξιολόγηση είναι η εξής:

```
accuracy after training on validation set: 73.3333%
>> |
```


Confusion matrix- πίνακα σύγχυσης

Ο πίνακας σύγχυσης (confusion matrix) είναι ένας τρόπος αξιολόγησης της απόδοσης ενός ταξινομητή σε προβλήματα ταξινόμησης. Απεικονίζει τον αριθμό των πραγματικών και προβλεπόμενων κλάσεων για κάθε κατηγορία του προβλήματος.

Ο πίνακας σύγχυσης παρέχει μια πληρέστερη εικόνα της απόδοσης του ταξινομητή, καθώς μπορούμε να δούμε πόσες προβλέψεις ήταν σωστές για κάθε κλάση και πόσες προβλέψεις ήταν λανθασμένες. Από τον πίνακα σύγχυσης, μπορούμε να υπολογίσουμε μετρικές αξιολόγησης όπως η ακρίβεια (accuracy), η ανάκληση (recall), η ακρίβεια (precision) και το F1-score για κάθε κλάση.



Από τον παραπάνω πίνακα σύγχυσης παρατηρούμε ότι το ταξινομητής μπορεί να αναγνωρίσει με μεγάλη ακρίβεια το γλυκά γαλακτομπούρεκο (86,7% ακρίβεια) αλλά δυσκολεύεται να ταξινομή τους λουκουμάδες όπου φαίνεται σε μερικές περιπτώσεις εικόνων από λουκουμάδες να τους ταξινομή ως μπακλαβάς. Το ίδιο φαινόμενο σε μικρότερο βαθμό εμφανίζεται και στο γλυκό μπακλαβάς.

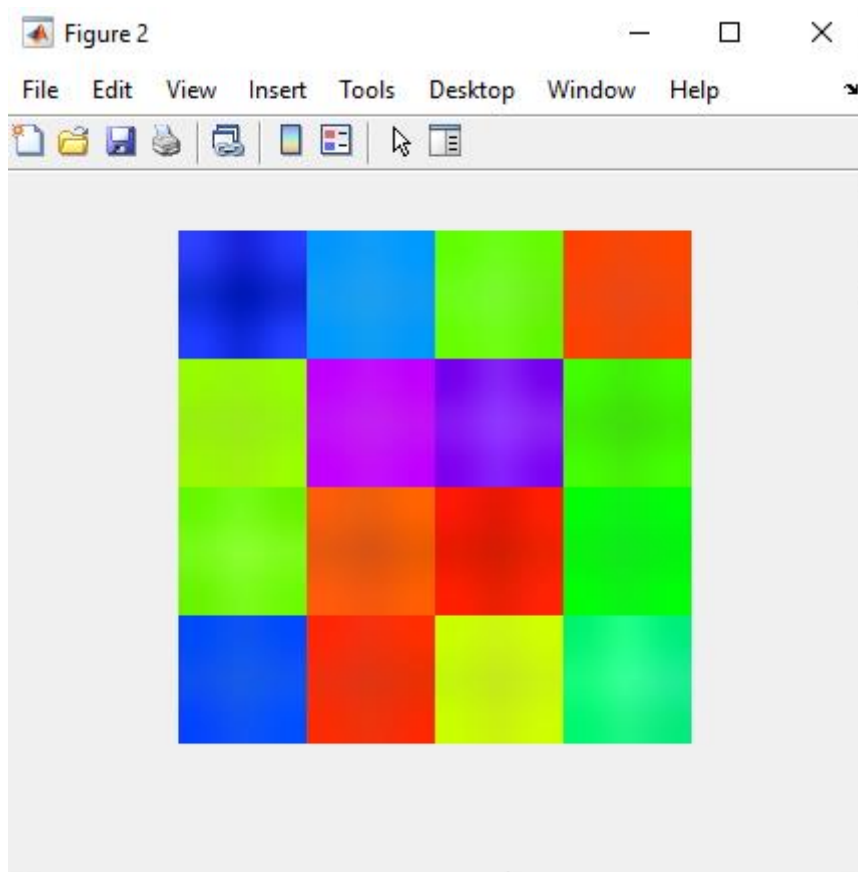
Ανίχνευση χαρακτηριστικών σε κάθε επίπεδο συνέλιξης

Τα συνελκτικά νευρωνικά δίκτυα χρησιμοποιούν χαρακτηριστικά για την ταξινόμηση εικόνων. Το δίκτυο μαθαίνει μόνο του αυτά τα χαρακτηριστικά κατά τη διάρκεια της εκπαιδευτικής διαδικασίας. Τι μαθαίνει το δίκτυο κατά τη διάρκεια της εκπαίδευσης είναι μερικές φορές ασαφές. Ωστόσο, θα απεικονίσουμε τα χαρακτηριστικά που έχει μάθει το δίκτυό μας.

Ανίχνευση χαρακτηριστικών, επίπεδο *Convolution 1*

Από επίπεδο συνέλιξης με 16 φίλτρα μεγέθους [3,3], και γέμισμα padding 0

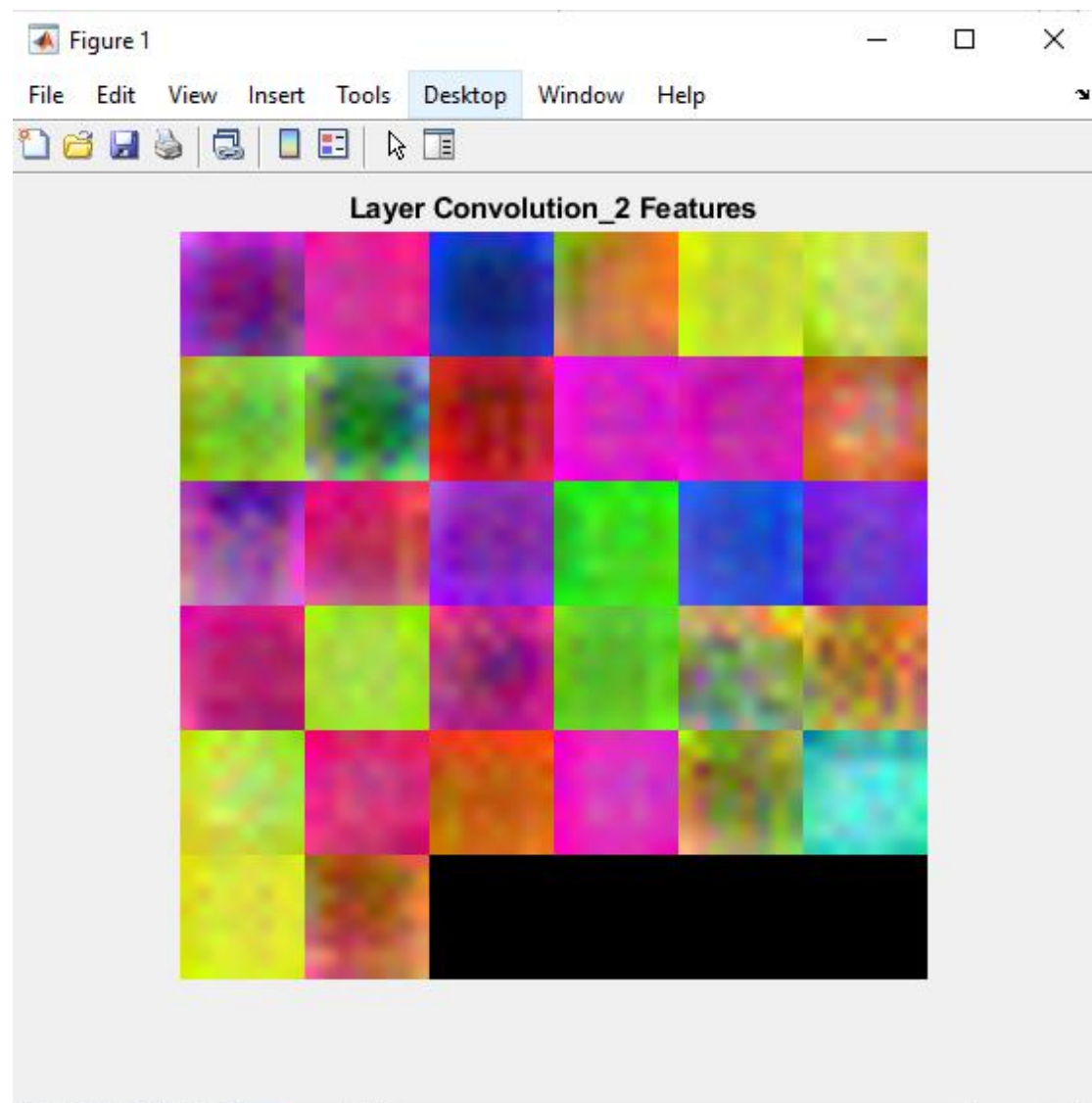
Αυτές οι εικόνες περιέχουν ως επί το χρώματα, γεγονός που υποδεικνύει ότι τα φίλτρα στο επίπεδο 'conv1' είναι ανιχνευτές χρώματος.



Ανίχνευση χαρακτηριστικών, επίπεδο *Convolution 2*

Από επίπεδο συνέλιξης με 32 φίλτρα μεγέθους [3,3], και γέμισμα padding 0

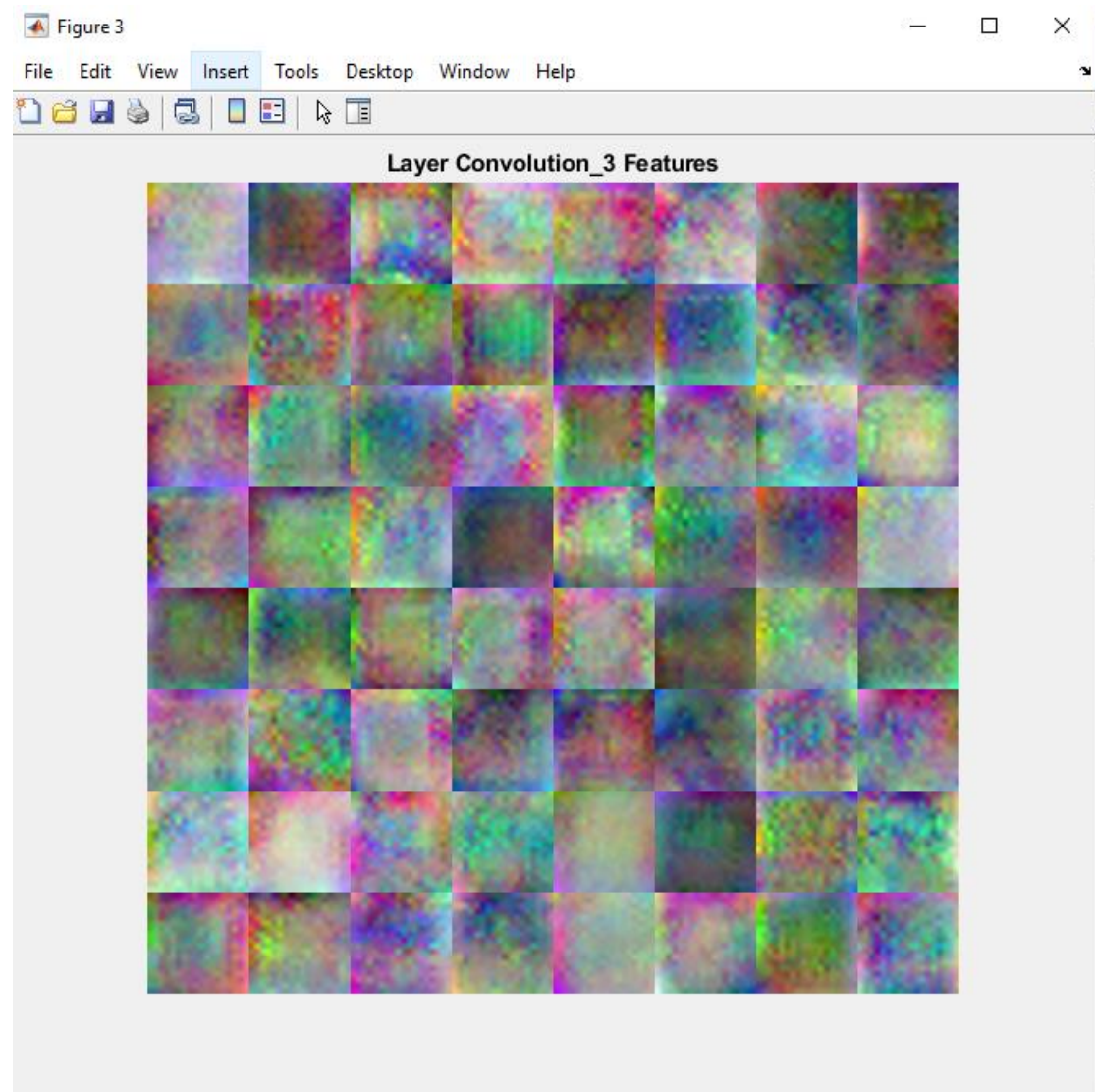
Αυτές οι εικόνες περιέχουν ως επί το πλείστον άκρες και χρώματα, γεγονός που δείχνει ότι τα φίλτρα στο στρώμα είναι ανιχνευτές ακμών και φίλτρα χρώματος. Τα φίλτρα για αυτό το επίπεδο ανιχνεύουν πιο πολύπλοκα μοτίβα από το πρώτο συνελικτικό επίπεδο.



Ανίχνευση χαρακτηριστικών, επίπεδο *Convolution 3*

Από επίπεδο συνέλιξης με 64 φίλτρα μεγέθους [3,3], και γέμισμα padding 0

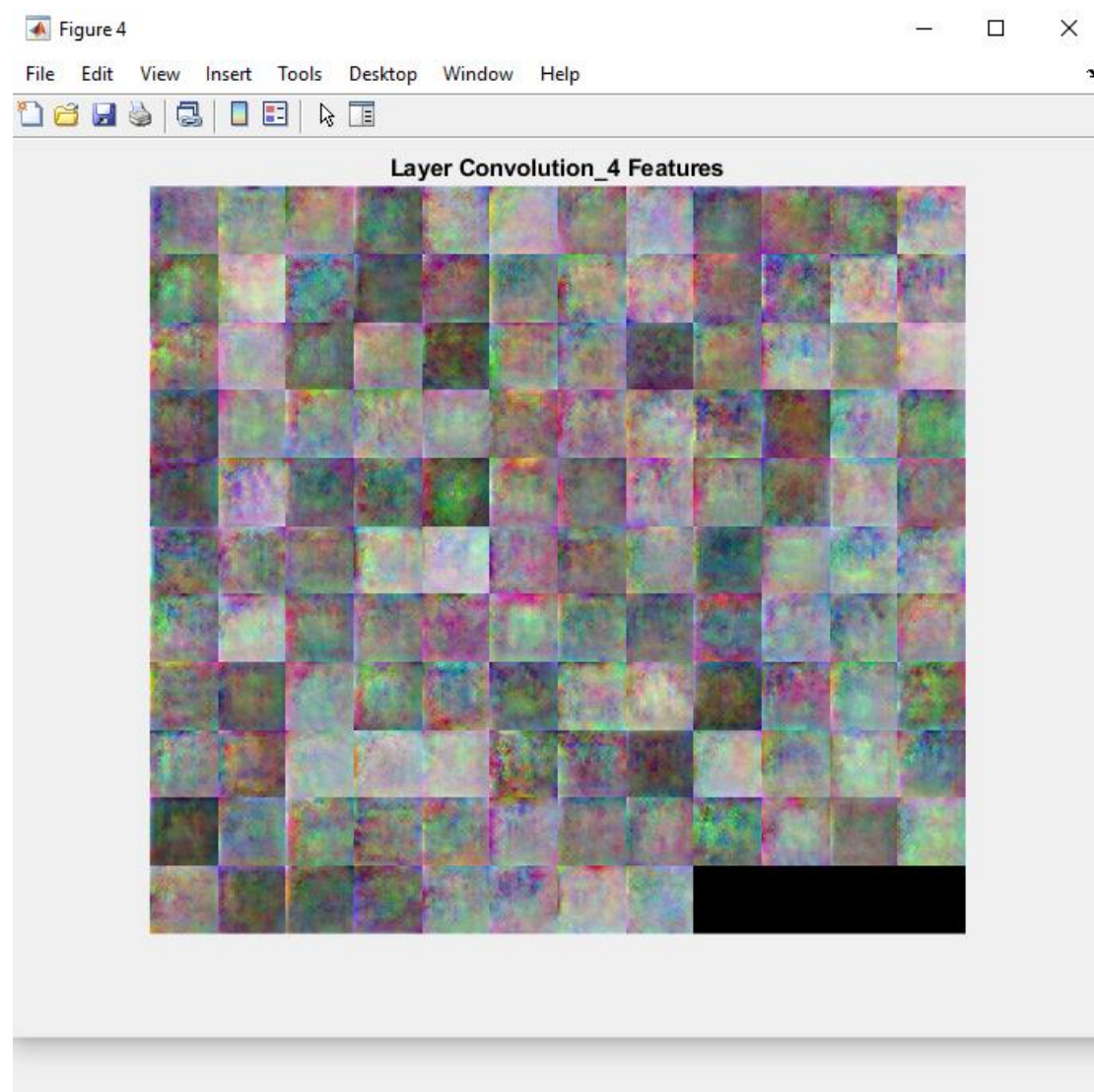
Παρατηρούμε ότι τα επίπεδα που βρίσκονται βαθύτερα στο δίκτυο παράγουν πιο λεπτομερή φίλτρα που έχουν μάθει πολύπλοκα μοτίβα και υφές.



Ανίχνευση χαρακτηριστικών, επίπεδο Convolution 4

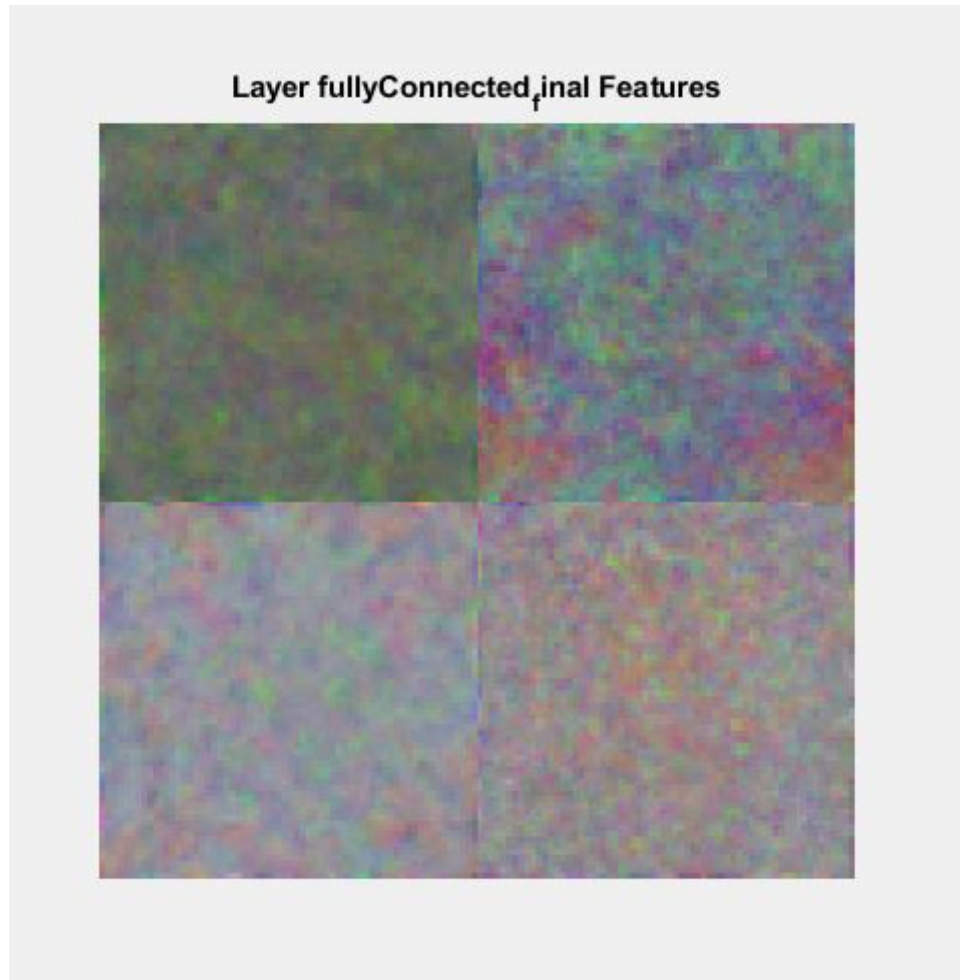
Από επίπεδο συνέλιξης με 128 φίλτρα μεγέθους [3,3], και γέμισμα padding 0

Ομοίως και εδώ παρατηρούμε ότι τα επίπεδα που βρίσκονται ακόμη πιο βαθιά στο δίκτυο παράγουν πιο λεπτομερή φίλτρα που έχουν μάθει πολύπλοκα μοτίβα και υφές.



Last layer extracted / discovered features

Δημιουργούνται λεπτομερείς εικόνες που ενεργοποιούν έντονα τις κλάσεις των γλυκών.



Συμπεράσματα – Προτάσεις για το μέλλον

Στη συγκεκριμένη εργασία ασχοληθήκαμε με την δημιουργία και την εκπαίδευση ταξινομητή για την εκπαίδευσή του στην αναγνώριση ελληνικών γλυκών. Η ακρίβεια του έφτασε συνολικά για όλες τις κλάσεις γλυκών στο 73,3%. Ωστόσο παρατηρήθηκε ότι κατά την αξιολόγησή του σε άγνωστες εικόνες φάνηκε να ταξινομεί λανθασμένα σε άλλες κατηγορίες γλυκών τα εκάστοτε γλυκά προς αξιολόγηση. Προτάσεις για να βελτιωθεί περισσότερο η ακρίβεια του ταξινομητή στο μέλλον είναι οι εξής:

- Πιο πολύπλοκο νευρωνικό δίκτυο
- Περισσότερες εικόνες γλυκών
- Επεξεργασία και εμπλουτισμός των ήδη υπάρχουσών εικόνων (image augmentation)