# Convolution Neural Network Vegetable Image classification

**Aarcha JayachandranNairPrasanna kumari** JYCRCH97S51Z222F@STUDIUM.UNICT.IT
**Monci Mamachan**[1] MMCMNC96R60Z222S@STUDIUM.UNICT.IT

## 1. Model Description

CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. One major advantage of using CNNs over NNs is that you do not need to flatten the input images to 1D as they are capable of working with image data in 2D. This helps in retaining the "spatial" properties of images. The deep model that which describe here consist of that consists of Convolutional Layers, to extract and recognize complex features, and by one Fully Connected Layer and a Classifier in order to perform an Image classification task for a set of Initially, input data flows trough the convolutional layers, to exploit image information. Then the extracted features are provided to the dense layer and to the classifier, which predicts the class result among one of the 15 possible classes.

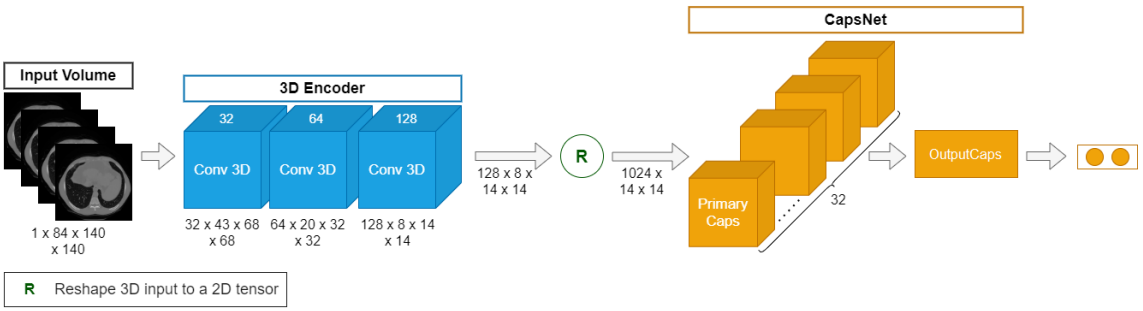


Figure 1: A simple representation of the proposed model.

At the beginning, it presents two convolutional layers, to process the input volume. The two convolutional layers have respectively as output 8, and 15 channels. The first layer has a kernel size of 3 applied with a stride of 1 in each dimension, followed by a ReLU activation function and by a layer of Batch Normalization. The second layer has a kernel size of 3 applied with a stride of 1 in each dimension, followed by a ReLU activation function and by a layer of Batch Normalization. Moreover, in this second layer, is applied also Max Pooling to reduce the size of the feature map for faster computations, keeping most important information. Features produced by these convolutional layers are then flattened and provided as an input to the Fully Connected layer. The dense layer has 1024 output features, taking as input 3600 features (the output of the convolutional layers). Finally,

the classifier has an output size of 15, that corresponds to the number of the classes of the dataset. Indeed, We will use the Cross Entropy as Loss Function, so the output will be a probability vector, meaning it represents predicted probabilities of all classes, summing up to 1. This method combines the Softmax activation function principle and the negative log likelihood loss.

## 2. Dataset

The initial experiment is done with 15 types of common vegetables that are found throughout the world. The vegetables that are chosen for the experimentation are- bean, bitter gourd, bottle gourd, brinjal, broccoli, cabbage, capsicum, carrot, cauliflower, cucumber, papaya, potato, pumpkin, radish and tomato. A total of 21000 images from 15 classes are used where each class contains 1400 images of size 224×224 and in *.JPEG format. The data set split 70This data set contains three folders:

- train (15000 images)
- test (3000 images)
- validation (3000 images)

each of the above folders contains sub folders for different vegetables where in the images for respective vegetables are present.

## 3. Training procedure

We haven't done any data augmentation, Here we trained a deep network based on the sequence of 4 Every convolutional layer follows a ReLU activation function, a Batch Normalization and, except for the first layer, a MaxPooling. Then we tried different attempts, removing one layer for each attempt. We tried 4 models,starting with 4 convolutional neural network to single convolutional neural network. Every model has one dense layer and a classifier. To compute the loss we apply as criterion the Cross Entropy Loss, that is the one used in classification models training:

1 n Xn i=1 (yi log( ˆyi) + (1 yi) log(1 yˆi)) (1)

where yi is the true output and hatyi the predicted one. All the models are trained from scratch for 5 epochs. Batch size is set to 32 for all the models. Stochastic Gradient Descent is chosen as optimizer algorithm using a learning rate of 0.01.

Max pooling The most popular form of pooling operation is max pooling, which extracts patches from the input feature maps, outputs the maximum value in each patch, and discards all the other values (Fig. 6). A max pooling with a filter of size 2×2 with a stride of 2 is commonly used in practice. This downsamples the in-plane dimension of feature maps by a factor of 2. Unlike height and width, the depth dimension of feature maps remains unchanged.

## 4. Experimental Results

This section presents the result of the classification accuracy obtained using CNN algorithm on various standard data sets.The results are presented using the classification accuracy in percentage for within train data and test data separately.Along with the classification accuracy percentage values

| Model | Train Accuracy | Validation Accuracy | Test Accuracy |
|---|---|---|---|
| – + Layer 1 | 99.41% | 93.65% | 93.68% |
| – + Layer 2 | 98.88% | 95.31% | 95.48% |
| – + Layer 3 | 97.08% | 95.88 % | 95.51% |
| – + Layer 4 | 94.46% | 94.58 % | 94.48% |

Table 1: Test performance of the models when using late arterial phase input data, delayed phase acquisitions or both. Best Network Model is with 3 layers.

## 5. Confusion Matrix

Figure 2: confusion matrix.