

Firsteable i want to thanks our professor Said benhlma for his tremendous help without him this project cannot see light like this. Also huge thanks to the jury ..... for evaluate this work . So this project's title is ML implementation ( NN) and the planning of this project is as follows :

### **Planning**

First i will start with introduction to the topic. Secondly we will talk about our first application Handwritting digits Recognition. We will show you

The dataset and libraries we used, then the split and preprocess of data. Furthermore mention the build of CNN model with keras. Then we'll make predictions.

Additionally we will explain our second application Control Media player with hand gestures we'll talk about data ressources, data size and data preprocessing.

Then building model with Tensorflow and Keras API, we'll define Hyperparameters and the architecture of the model. After this compile and fit Keras model .

Last Controlling media player and Finally we'll sum up with a conclusion.

### **Introduction**

Machine learning is one of the most exciting recent technologies . Every time you use a web search engine like Google to search the internet, one of the reasons that works so well is because a learning algorithm, one implemented by Google or Microsoft, has learned how to rank web pages.

Every time you use Facebook or Apple's photo typing application and it recognizes your friends' photos, that's also machine learning.

So machine learning is a : Subset of AI techniques which is a technique to achieve Ai through algorithms trained with data

Ai : Technique that enables a machine to mimic human behavior

Dl is .... Subset of ML inspired by the structure of the human brain

In terms of deep learning this structure is called an articificial nn

### **Appl**

Handwritting digits recognition has gained so much popularity from the aspiring beginner of machine learning and deep learning to an expert who has been practicing for years. Developing such a system includes a machine to understand and classify the images of handwritting digits as 10 digits (0-9).

### **NEcessary libraries**

Numpy offers speedy computation and execution of complicated functions working on arrays.

Opencv to provide a common infrastructure for computer vision applications and accelerate the use of machine learning.

Tensorflow is one of the best library available for working with machine learning on python offered by google. TensorFlow makes ML model building easy for beginners and professionals alike.

Keras was soon supported in tensorflow to build deep learning models

Any object in python can be pickled to that it can be saved on disk

Tkinter provides a fast and easy way to create GUI applications

### **build the model using cnn**

This is the model of handwritten digit recognition using convolutional neural networks

These images have been scaled down to 32 by 32 pixels. The images are in gray scale to reduce the amount of information. So with 32 by 32 pixels, 1024 bytes are needed to store the entire image. Rather than looking at an entire image at once to find certain features it can be more effective to look at smaller portions of the image. So ConvNet architectures allows us to encode certain properties into the architecture. Each input image will pass it through a series of convolution layers with filters (Kernels), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. So we take a filter of a size 5x5 and we move this across the image from top left to bottom right. For each point on the image, a value is calculated based on the filter using a convolution operation. When building the network, we randomly specify values for the filters, which then continuously update themselves as the network is trained. We can then do a lot of things, such as adding more filtering layers and creating more feature maps, which become more and more abstract as we create a deeper CNN. We can also use pooling layers in order to reduce the size of the representation. max pooling is used because we want to find the outliers. In order to build deep neural networks, one modification to the basic convolutional operation that one needs is padding. Padding essentially makes the feature maps produced by the filter kernels, so we choose valid padding which keeps only valid part of the image and drop the part of the image where the filter did not fit. Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. ReLU's purpose is to introduce non-linearity in our ConvNet. Since, the real world data would want our ConvNet to learn would be non-negative linear values. So we add as many convolutional layers until satisfied; instead of having 1024 features, thanks to CNN we have only 480 features. Then The layer we call as FC layer, we flattened our matrix into vector and feed it into a fully connected layer like a neural network, which we choose 500 nodes. there is no hard and fast rule to choose the number of hidden-layers or nodes .

### **implementation**

Then we are going to create our first layer which is model.add and we are going to add a convolution layer, and we will define the number of filters and the size of the filters, here we are using bigger filters by the size 5 by 5 and later on we are going to reduce the size to 3 by 3, then we are going to input the shape with the size of (32, 32, 1) and finally the activation function is a rectified linear unit 'relu'. Next we are going to add another one of these convolutional layers and this time we will keep everything pretty much the same but we don't need to define the dimension again. Then we are going to add a max pooling layer. We will define the pool size, so the pool size is equals to (2, 2). Then we will add another two convolutional layers, for both of them this time we are going to decrease the number of filters by half and we change the size of the filter number this time we use smaller filters (3x3). Again we add the Maxpooling and then we add our first dropout layer, so 0.5 means 50% of the neuron excluded from each update cycle. So we are going to write the flatten layer then we add our dense layer which we define the number of nodes we choose 500 and then the activation we can keep it relu. Then we will add another dropout layer that can helping to reduce the overfitting it's making it more generic. At the end we are going to add another dense layer which will be our final layer so we will have the number of classes and this time we will use the softmax function. This is our model, all we need to do now is to compile, we have to mention the optimizer which is

Adam and the learning rate for it is 0.001 and the loss is categorical crossentropy, finally we are going to return our model.

## **app2**

### **libraries:**

with pyautogui you can control the mouse and the keyboard to automate interactions with other applications.

### **model:**

This is the model of hand gestures using CNN

These images have been scaled down to 128 by 128 pixel image with 3 color channels there are around 49152 weights that must be trained.

So we need ConvNet to reduce the amount of parameters in the network to speed up the computation as well as to make some of the features

that detects signals more robust. Each input image will pass it through a series of convolution layers with filters (Kernels), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. So we choose a filter of a size 3x3 in convolutional layers and 2x2 in max pooling layers to reduce the size of the representation by divided the size in half. so we choose 32 filters

after we worked with 64 filters, in the first and second layer filters capture patterns like edges, corners, dots etc. In the last layer we combine those

patterns to make bigger patterns. Like combine edges to make squares, circle etc. so we choose valid padding. And the stride is 1 then we move the filters

to 1 pixel at a time. So we add as many convolutional layers until satisfied; instead of having 49152 features, we have now 12544 features. Then we flattened

our matrix into vector and feed it into a fully connected layer which we choose 4096 nodes in first fully connected layer and 2048 nodes in second fully

connected layer, then 8 nodes in the output layer. So we decrease the number of neurons in FC layer for decreasing the number of parameters.

### **implementation**

The difference between the previous and current model is the number of filters, size of filters, number of nodes, and number of layers furthermore the percentage of dropout.

### **conclusion**

We are able to program a handwritten digit recognizing and controlling media player with hand gesture, using neural networks to learn from dataset how to recognize handwritten digits, and how to recognize hand gestures to control media player, then apply what it learned on a testing set with a precision greater than 98%.

### **perspective**

For the second project we can deploy it in a micro controller like raspberry pi with an external camera to control a TV or a self driving car that's make it more applicated.

For two applications are available for anyone who wants to test them in computer, you can find this project on own github.