Text

Description automatically generated

**Fall 2021**

**Intro to. artificial intelligence**

**CSC 4301**

**Project Report**

**Subject: Handwritten mathematical symbols identification**

**Groups:**

**Driss Jaidi and Mohammed Chaouni**

**Supervised by: Dr. Tajjeeddine Rachidi**

**Table Of contents:**

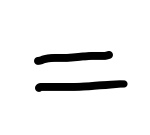
1. **Introduction**
2. **Technologies used description:**
3. **How the program works:**
4. **How we can apply this program in a real-life application:**
5. **Conclusion:**
6. **Project Description:**

In this project, we developed a program that takes images of handwritten mathematical symbols and tells us what they are. The program can support any handwritten symbol if there is enough training data meaning more than 100 images at least.

Supported symbols:

Addition, almost equal, alpha, belongs to, beta, dot, different than, division, does not belong to, equal, equivalence, fi, for all elements, integral, intersection, left-bracket, mu, omega, pi, percentage/modulo, R numbers set, set of integers, sigma, subtraction, union, and equivalence.

**Une image contenant aéronef

Description générée automatiquement**

1. **Technologies used description:**

**imported utilities:**

Une image contenant texte

Description générée automatiquement

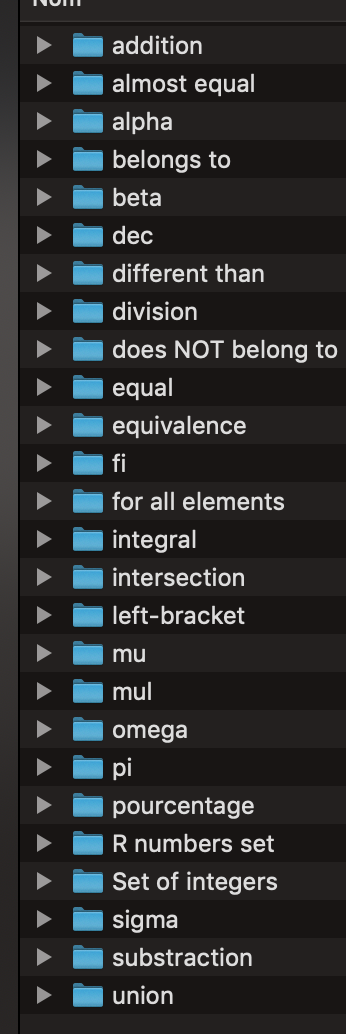
We used TensorFlow keras and NumPy to load images into arrays and build matrices

and sklearn to generate metrics that helped split data and generate a good CNN.

1. **How the program works:**

* Import the dataset file directory that must contain each symbol we want labelled as the output name we want.

Une image contenant texte, extérieur, signe

Description générée automatiquement

* Test if the file is fetched and does exist

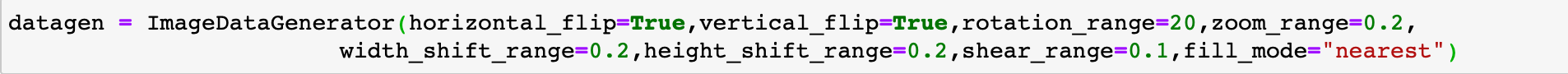
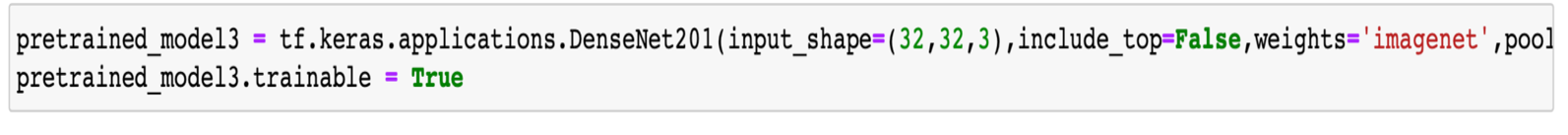
Une image contenant texte

Description générée automatiquement

* Zip the files names into a dictionary called reverse mapping that will be used later to give the predictions of new images other than training and dataset/ the mapper function is the most important here since it returns that value: Une image contenant texte

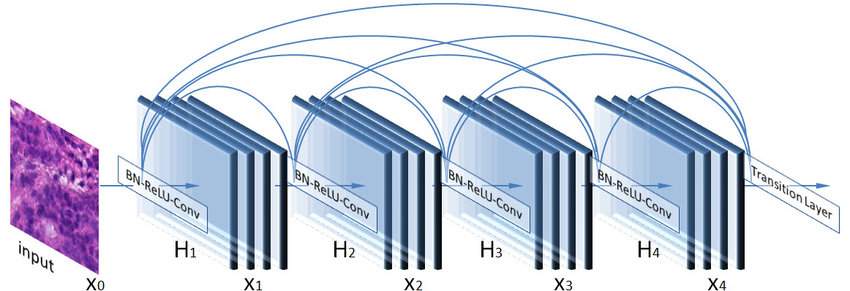
  Description générée automatiquement
* Split the training and datasets using an optimal value based on error and retry until the end results are above 90% accuracy at the end.Une image contenant texte

  Description générée automatiquement
* The training sets (datasets) and test sets (testsets) are splits into labels and values, then those labels are mapped into a binary matrix using to categorical () as shown below. Then they are split into train x and test x containing the shapes, imgages, and their labels and train y test y here the test size is 20% or 0.2 that is why we split them using 80% or 4/5 for training before and 1/5 for testing.Une image contenant texte

  Description générée automatiquement
* We use the ImageDataGenerator that will use existing data to generate condition to avoid overfitting like the rotation range, zoom range, etc. as seen in the code snippet below. This allows to avoid giving useless data and saves a lot of effort by letting ai help us in this problem. Instead of building it from scratch. Because collecting the existing images is already time consuming so variations manually would have even taken much longer.
* Instantiate the Densenet201 architecture which will be used to build the neural network of this AI project:

We also set the model as trainable “true” because “false” increases the error rate even though it had more than 80% accuracy too but too linear and an 11% advantage is much better.

How it works:



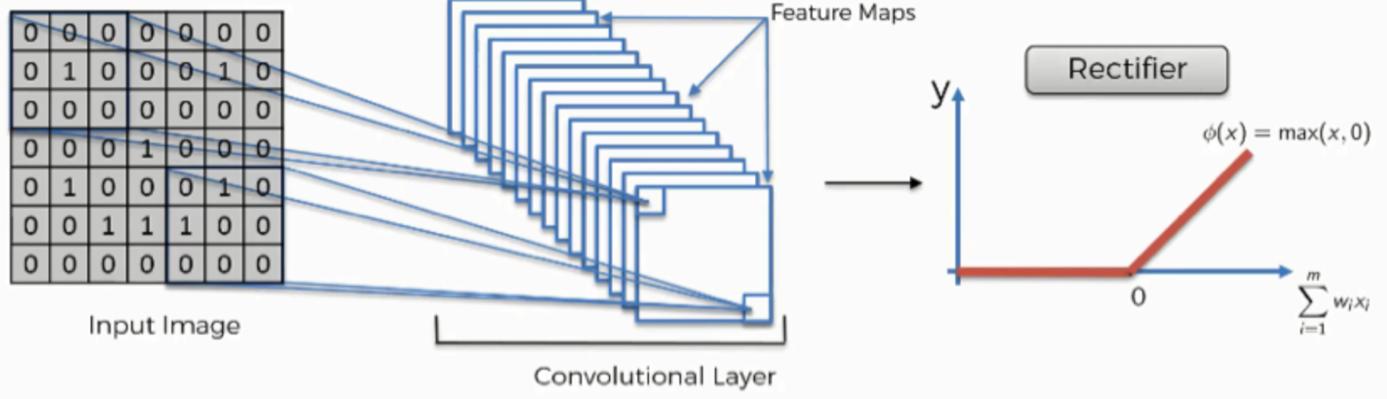
Source : <https://www.researchgate.net/figure/DenseNet201-Block-Architecture_fig1_333827777>

* We set the denseNet201 as 128 nodes in the middle layers and 26 nodes in the output layer. As it has 201 layers according to its definition it takes a lot of GPU and CPU power to compile. We set the middle layers in such a way they must be connected to the value of the input pretrained model which affects the bias. As the activation function for the black box layers is “**relu**” which is the rectified linear activation function which helps a lot in the CNN “convolutional neural network” to avoid exponential growth of matrix calculations and value and only keep Positive values. Densenet201 is also much better than ResNet as it decreases bottlenecks and reduces unnecessary memory usage.

Une image contenant texte

Description générée automatiquement

How it works:



Source:

<https://mohameddhaoui.github.io/deeplearning/CNN_tuto11/>

while the “**SoftMax”** activation gives a probability distribution from the neural network selected values which decides what value will be selected last.

* Compile the model as to allocate memory for it:



Instead of gradient decent optimization algorithm we used the Adam algorithm as it uses less memory and is more efficient in computations and it does everything behind the scenes. The “categorical cross entropy” compares the predicted model to the test data probabilities and gives the loss probability. The accuracy describes how accurate the model values are.

* We fit the model with what we gathered from training and testing data through 10 iterations the set is limited to 10 to avoid overfitting and under fitting the “accuracy” metric is the most important since we got so many mistakes despite having 0.9> value accuracy so with the below values we found the most optimal version with minimal errors meaning one or two errors out of 10:

Une image contenant table

Description générée automatiquement

* We set the predictions and ground meaning basis using the testy which helps generate the classification report as shown below and the confusion matrix I github

Une image contenant table

Description générée automatiquement

* Finally test with foreign data:

Example:

Une image contenant texte

Description générée automatiquement

1. **How we can apply this program in a real-life application:**

* Whenever a user opens their phone keyboard the screen records the small portion where the user will handwrite then 3s after the user lets go off the screen the program generates the written mathematical symbol. Convert the final frame of the video into an image and set it as an input for the neural network.
* Be used as a basis ground to make a bigger program that handwritten recognizes mathematical expressions and convert them to text. (From paper images that will have to be converted into very white images with the writing set as a black color to make it easier for the NN)

1. **Conclusion:**

All in all, this was a great project to get introduced to neural networks and how data is very important since it is more important than the neural network itself since a NN is useless without data. And the neural network is really a great way to. simulate a small capability of the human brain to recognize things. Since there are so many things, we can do with it.