Homework Marking System

Arthur LANG
School of Engineering
Jönköping University
Jönköping, Sweden
laar20vx@student.ju.se

Madalina Aldea
School of Engineering
Jönköping University
Jönköping, Sweden
alma20px@student.ju.se

***Abstract*—**Homework are an important part of the learning process, helping the student to better fix knowledge. But if homework give work to students, it is also a certain amount of work for the teachers, who have to correct all the submissions. This is estimated around 3 hours of work every week in average [1].We think this time can be reduce by automated the process of correction. This paper proposes a method for automatically correct homework. For simplification reason, we decided to focus on submission in primary schools.

Keywords— Homework marking, Machine learning, Computer vision,

I. INTRODUCTION

It is clear that an AI cannot replace a teacher. Many of the attributes that make good teachers are the very things that AI or other technology fails to emulate: inspiring students, building positive school and class climates, resolving conflicts, creating connection and belonging, seeing the world from the perspective of individual students, and mentoring and coaching students [1].
But teachers spend time on subject that can be optimized, as the homework evaluation. This time gain is estimated a bit less than 50% with automation of the correction [1].

II. THE PROBLEM

The risk when you talk about homework correction, is the diversity of format, subjects, exercises. We had to decide a scope of our project and focus on a simple exercise. The first thing that come to mind was the math. Indeed, a math exercise is, most of the time, true or false. It removes all the interpretation part that could be present in a literature exercise for example. Moreover, we wanted to focus only on the result part for the moment. It could be look oversimplified, but our main fear was the difficulty to recognize children handwriting that is more various and unsure than adults one.

That why our focus with this project was the children handwritten recognition through assignment correction. Now we have our frame, we can start to work.

III. APPROACH

We took a certain amount of time to define exactly the workflow that will be follow by the teacher. We had several problematics: first, we needed to have a version of the homework with the correct answer, to keep it as the reference and correct the rest. Secondly, we do not want the teacher to selected for every children's submission the area of each question, in this case it do not save time but just move it to another task. On the other hand, we were limited in term of time to add the search of the answer by the algorithm.

We finally agreed on the following approach: the teacher hand in a paper with the correct answers and uses the program to mark the position of each answer (we called that Regions Of Interest, ROI). Our idea here was to limit the number of data to treat: despite of interpreting the subject and try to guess the answers, the user can directly specify where the answers are. Moreover, when a teacher gives an exercise to his children, all the subjects are the same, so the children answers are in the same area.

The ROI given by the teacher are then crop on each submission. We need then to identify the digit in the teacher paper and the one in the children work. For that, we will need to train a model for handwritten recognition.

At the end of the program, we wanted to give a nice feedback for the teacher, by marking with a green box the correct answer and with an orange box the wrong one. We also wanted to output the digit recognize by our algorithm, to be able to check if the recognition went well or not. In the future, this feature can be removed for the correct answer, but will be kept for the wrong one.

IV. SOLVING THE PROBLEM

A. Choosen technologies.

We first used python as the language for several reasons: a lot of machine learning libraries and tools are implemented in python and simply, it was the language that was the best master between us two.

In complement of python, we used numpy library, that allow a great an optimized gestion of array in python.
The model was built using Keras. Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow[2]. Keras has an user-friendly and intuitive design which makes it easy to use. Also, each model can take advantages of pre-built function for compiling, fitting and evaluating it. Keras also provides methods for saving or loading a model, meaning that a model can be trained just once and use it for many times without needing to train it again. Models in Keras are constituted by a sequence of layers, added one at a time. Keras can be used in a wide range of areas, some examples being facial recognition, object recognition, speech recognition, grammar learning, sign language translation and many others.

Because we were manipulating pictures, we used also OpenCV. This library is famous and really complete about image manipulation. It is implemented also in python, that fit with our technical stack and is compatible with numpy. Note that openCV also integrated some machine learning tools.

B. *Dataset*

We spent a lot of time searching for a good dataset, but unfortunately, we were not able to find one that fit exactly our problem. That is why we first used the Mnist dataset. It is the "hello world" of handwriting recognition, a dataset composes of 70.000 (60.000 for training and 10.000 for test) pictures with digits between 0 and 9.

To complete our dataset, we took contact with a teacher who helped us by handing in to student some worksheets on which children wrote digits and solved equations. On the first task, the students were supposed to write 5 times each number between 0 and 10 in this way we obtained a dataset which was used for fine tuning the model. In total, we had 340 samples of digits (the worksheets were scanned and the quality for some digits was poor, making them impossible to use). The worksheets needed to be processes in order to be able to use the information, each digit was cropped and labeled accordingly. After that, the images were processed following the mnist digits format.

At a moment, we also used an extended version of Mnist, with number from 0 to 99 but didn't keep it for too long.

C. *Choosing a model*

The model was built using modules provided by Keras. We tried several models. Following some knowledge [3], we try to fit our hyperparameters to our dataset and problem. The first training sessions were run on the following model architectures:

a) The first model

```
model = Sequential([
    Conv2D(filters = 32, kernel_size=(3,3), activation='relu', input_shape=(28, 28, 1)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPool2D(2, 2),
    Dropout(0.25),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(100, activation='softmax'),
```

Fig. 1. Architecture of the first model

In Keras, a model is a sequence of layers, added one by one. The convolutional layers and the MaxPool layer are used for feature extraction and for downsampling the input by taking the maximum value over a defined window. The dropout layers are used to prevent over fitting, flatten layer is used for reshaping the data for the dense layers. Dense implements the operation: output = activation(dot(input, kernel) + bias) where activation is the element-wise activation function passed as the activation argument, kernel is a weights matrix created by the layer, and bias is a bias vector created by the layer [4].

Using the above model, we were able to obtain good accuracy (over 99%) in the training and evaluation phases. Also, this model was trained on the extended version of Mnist. We decided to change the model after several tries of corrections on two digits numbers and on digits written by children.

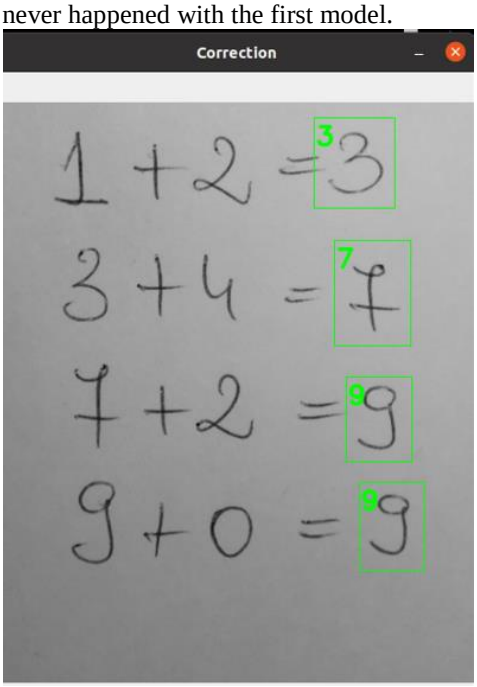
b) The second model (final model)

```
model = Sequential()
model.add(Conv2D(32,(3,3), strides=(1, 1),  activation="relu",input_shape = (28,28,1),data_format = "channels_last", use_bias = True))
model.add(Conv2D(32,(3,3), strides=(1, 1),  activation="relu", use_bias = True))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2,2)))
model.add(Dropout(0.2))

model.add(Conv2D(128,(3,3), strides=(1, 1),  activation="relu", use_bias = True))
model.add(Conv2D(128,(3,3), strides=(1, 1),  activation="relu", use_bias = True))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2,2)))
model.add(Dropout(0.2))

model.add(Flatten())
model.add(Dense(256,activation = "relu", use_bias = True))
model.add(Dropout(0.5))
model.add(Dense(11,activation = "softmax",use_bias = True))
```

Fig. 2. Architecture of the second model

Since on samples from children handwriting the performance was poor, the structure of the model was changed as shown above. Extraction of the features and down sampling are done twice using two blocks with convolutional layers and a MaxPooling2D module. Also, two new modules were integrated (BatchNormalization()) for speeding up the training and for avoiding internal covariate shift which appears because of the changes in network parameters during training phase.

Using this configuration, the accuracy had values between 99.3% to 99.4%, like the first model, but the performances on children's handwritten digits were better, few times, all the digits were correctly predicted, something that never happened with the first model.



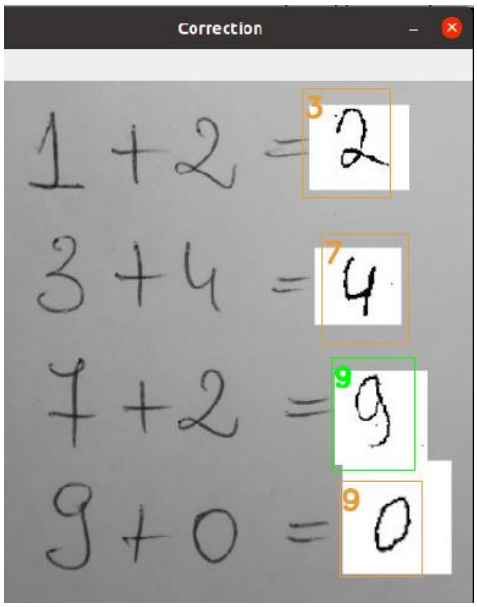


Fig. 3. Samples of correction

c) Evaluate the model

Concerning the evaluation of the model, first we used the accuracy indicator to know about the efficiency of our model. Accuracy is quite a good indicator in our case because the Mnist dataset is balanced. We did also some test from our own writing and writing we collected from children, to allow us to see how work the model in front of data he does not know. Even the data we collected from children was balanced: each child had the same subject and write the same number of digits. Even if we skipped few digits, less than ten, from our collection, it represents only 0.02% of the total dataset. It is too few to unbalance the dataset.

d) Training

We split the training part in several episodes. First, and it was our starting point of the project, we train a simple example with Mnist dataset. We had a good accuracy on the dataset, around 0.99 %. But when we tried it on some pictures we got from children, the result was worst. We did 4 tests, only 2 were good.

At this point, we decided to still keep this model, the time for us to develop the other tools around. (see Utilities around paragraph).

But we continue to think about the reasons of the good accuracy for a poor result. The obvious answer was because the dataset we trained with, is quite different from the pictures we had from children. For example, mnist dataset has only centered digit, that was not our case. Moreover, children's writings are more unsure, weirdly formed.

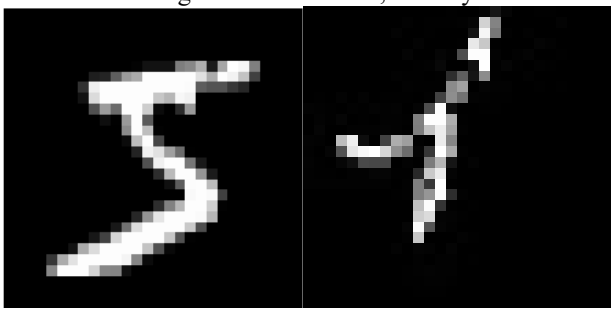


Fig. 4. Sample of digits : left –mnist sample, right-sample from digits written by children

Another problem we were aware of, is about compose number, like 12, 43 or 938. Our first track was to split a number like 12 in 1 and 2. Then we could identify each digit and merge them at the end. But after several research and trials, we couldn't separate them from a sure way. Our main problem was when the digits are too close, when they "touch" each other or when the shape of the digit is not perfectly defined. But we found a dataset, a sort of augmented mnist [5] to generate all the digits between 0 and 99. We did a training session, but the test was worse than before. We thought it was due to a huge number of features, more present than the first training. On the same 4 digits we tried before, we got 1 good digit and the wrong digits were vaguer (the first training mixed a 9 with a 3, that are quite similar, but the second mixed a 3 with 12 for example). We didn't go too far with this dataset and preferred to focus on improving the model for digit detection between 0 and 10.

We decided then, after some discussion with our teacher, to try a little custom dataset, with the digits we got from children, mixed with some examples from mnist repo. That took quite some time to crop several digits we get and label them.

We trained the model using the mnist dataset and retrain it using a small dataset containing digits written by children. This approach was the best we had, the accuracy was, like the previous one, more than 0.99, but the model being able to recognize correctly almost every digit from a worksheet.

Our model is not perfect, we sometimes have mismatched digits. Some reasons of this result could not be the model but the crop for the following reasons: the digit is not always center, not like the Mnist dataset, and we have to resize the crop in 28x28, so that could deform the digit.

e) Utilities around

Having a functional model was not our only goal, but we wanted to have a minimum program that can be used by a teacher or easily integrated in a Graphical User Interface.

Then we had the following architecture:

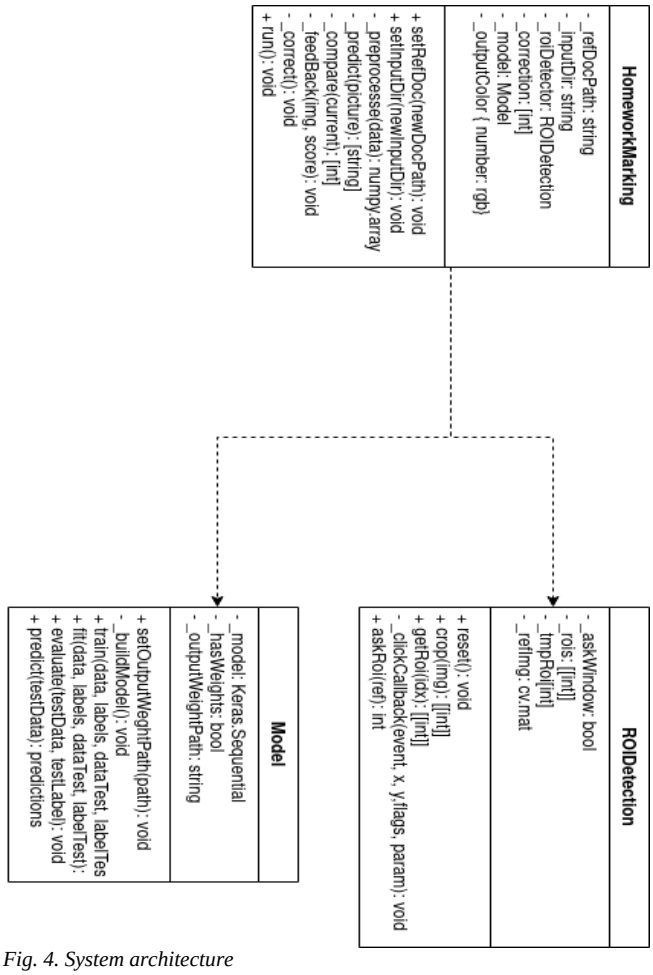


Fig. 4. System architecture

HomeworkMarking is the main object that link all the process. It is inspired by the Mediator Design Pattern [6], but still contain some logic, to avoid multiplication of objects.

ROIDetection is the object that manage all the Rois. Note that an ROI is defined as an array of 4 int, representing the coordinates x and y of the high left point and low right point of the ROI, that are a rectangle.

Model is an encapsulation of a keras model.

f) Disclosure of Contribution

Find bellow a table with all our task, how much time we spent on it.

TABLE I. DISCLOSURE OF CONTRIBUTION

Task / Member	Mădălina	Arthur
Research	15h	12h
Dataset work	12h	8h
Building Model	9h	10h
Training Model	20h	17h
Utilities development	15h	20h
Write the report	4h	6h
Documenting	1h	2h
Lab Sessions	4h	4h
Total	80h	80h

V. CONCLUSION

Identify children writing give us more challenges than excepted, and we conclude that we need a bigger dataset to have a real-life applicable solution. But we are still confident about the result we get and think that the bases we put on this project could be easily expends.

REFERENCES

[1] Bryant, J., Heitz, C., Sanghvi, S., & Wagle, D. (2020, December 04). How artificial intelligence will impact k-12 teachers. Retrieved March 21, 2021, from https://www.mckinsey.com/industries/public-and-social-sector/our-insights/how-artificial-intelligence-will-impact-k-12-teachers

[2] Team, K. (n.d.). Keras documentation: About Keras. Retrieved March 21, 2021, from https://keras.io/about/

[3] Hyper-Parameter Optimization: A Review of Algorithms and Applications Tong Yu, Hong Zhu

[4] Team, K. (n.d.). Keras documentation: Dense layer. Retrieved March 21, 2021, from https://keras.io/api/layers/core_layers/dense/

[5] Shaohua0116. (n.d.). Shaohua0116/multidigitmnist. Retrieved March 21, 2021, from https://github.com/shaohua0116/MultiDigitMNIST

[6] Mediator. (n.d.). Retrieved March 21, 2021, from https://refactoring.guru/design-patterns/mediator