**Aksara Jawa Image Classification Application**

**Group: DPS4-C**

**Members:**

1. Biyan Oscar
2. Ni Luh Ayu Sonia Ginasari
3. **Choice dataset Reason**

We chose this dataset because this dataset is of good quality and covers 20 basic characters of Javanese script.

Aksara Jawa, also known as Hanacaraka, Carakan, or Dentawyanjana, is one of Indonesia's traditional scripts that developed on the island of Java.

But for the use of Aksara Jawa at the present time is only limited as a regional symbol that is embedded in the names of the street, meeting buildings, government buildings, and others.

It is clear that the Aksara Jawa at the present time is not functioning enough, so that its presence is increasingly invisible. Aksara Jawa is considered outdated, ancient, and traditional. While the world of work is already using modern tools.

1. **Baseline CNN implementation**

Colabs code can be seen at:

<https://github.com/biyanoscar/bangkit-dps4c/blob/master/aksara_DPS4C_Baseline_CNN.ipynb>

Image size used: 224 x 224. We use relu activation and the number of filters are 3.

Neural Networks architectures:

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv2d (Conv2D) (None, 222, 222, 16) 448

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d (MaxPooling2D) (None, 111, 111, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_1 (Conv2D) (None, 109, 109, 32) 4640

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_1 (MaxPooling2 (None, 54, 54, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_2 (Conv2D) (None, 52, 52, 64) 18496

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_2 (MaxPooling2 (None, 26, 26, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten (Flatten) (None, 43264) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense (Dense) (None, 256) 11075840

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

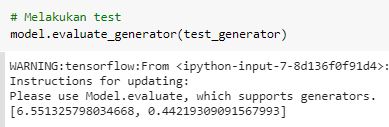
dense\_1 (Dense) (None, 20) 5140

=================================================================

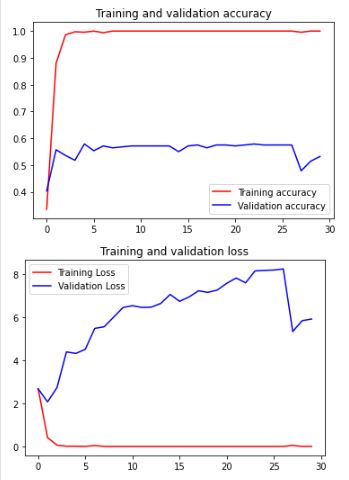
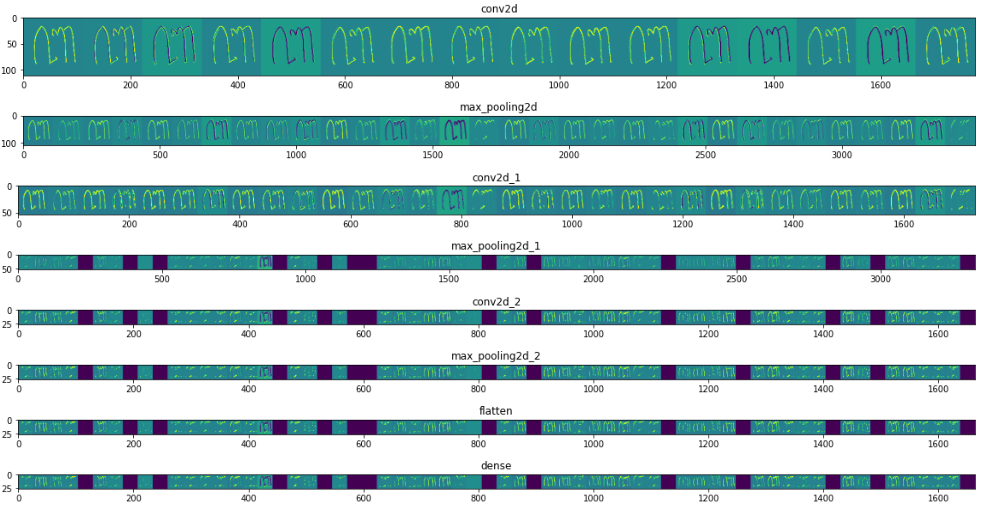
Total params: 11,104,564

Trainable params: 11,104,564

Non-trainable params: 0

It can be seen, the accuracy in the test dataset is low, the accuracy is around 44%.

In addition, from the accuracy curve of the model during training, it can be seen that there are symptoms of overfit in our baseline model.



In the activation layer plot, it seems like there are dead neurons, because some images do not appear good, they are just blank images after the second max pooling.

1. **Selected improvements**

The following is a list of improvements that we chose

1. Image Augmentation

Keras ImageDataGenerators is the tool used for generating more training data from the original data to avoid model overtting. It is conducted online by looping over in small batches during each optimizer iteration.(Gu, Shanqing et.al., 2019)

By using image augmentation, we generate new images for increasing the volume of the training dataset. Our image dataset is a small dataset, so it's more susceptible to overfit.

1. Using leaky ReLU as activation functions in convolutional neural network

Based on research paper, “we conclude on small dataset, Leaky ReLU and its variants are consistently better than ReLU in convolutional neural networks”.(Xu, Bing et.al., 2015)

1. Dropout Layer

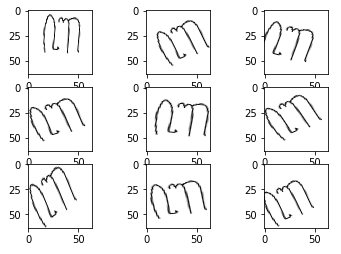
A dropout layer is used to avoid neuron interactions and learn more robust features in all training data(Gu, Shanqing et.al., 2019)

1. **Improved CNN Implementation**

We reduced the size of the training input image to 64x64 so that the model size was smaller. And then we use 3 techniques that we mentioned before.

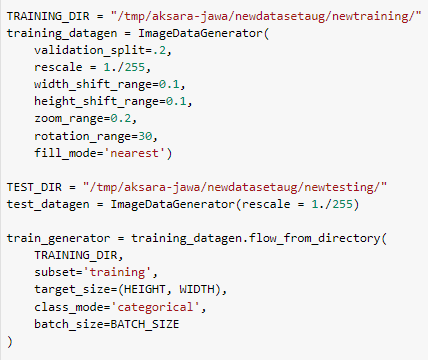
First we set up image augmentation.

We test the augmentation, to see which image augmentation is suitable for use. We use Image Data Generator to generate that.



So we use this image augmentation: width and height shift, zooming, rotation.

By Image Data Generator, we also split data into training and validation dataset with a ratio of 0.2.



Next we build a Deep Neural Network using keras. In the model, we add LeakyReLU and Dropout layer, so it overcomes the overfit problem.

This is the summary of the model architecture.

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv2d (Conv2D) (None, 62, 62, 16) 448

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

leaky\_re\_lu (LeakyReLU) (None, 62, 62, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d (MaxPooling2D) (None, 31, 31, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_1 (Conv2D) (None, 29, 29, 32) 4640

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

leaky\_re\_lu\_1 (LeakyReLU) (None, 29, 29, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_1 (MaxPooling2 (None, 14, 14, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_2 (Conv2D) (None, 12, 12, 64) 18496

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

leaky\_re\_lu\_2 (LeakyReLU) (None, 12, 12, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_2 (MaxPooling2 (None, 6, 6, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten (Flatten) (None, 2304) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense (Dense) (None, 128) 295040

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout (Dropout) (None, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

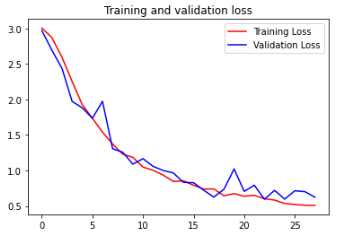
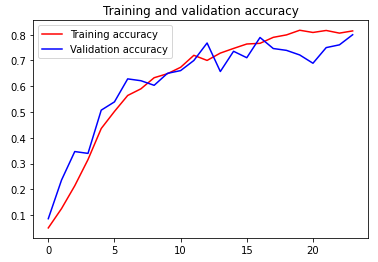
dense\_1 (Dense) (None, 20) 2580

=================================================================

Total params: 321,204

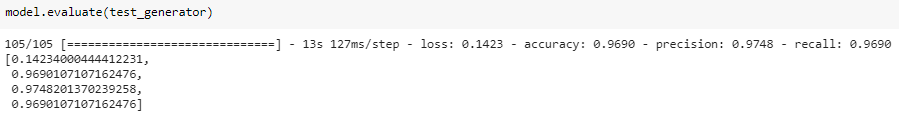
Trainable params: 321,204

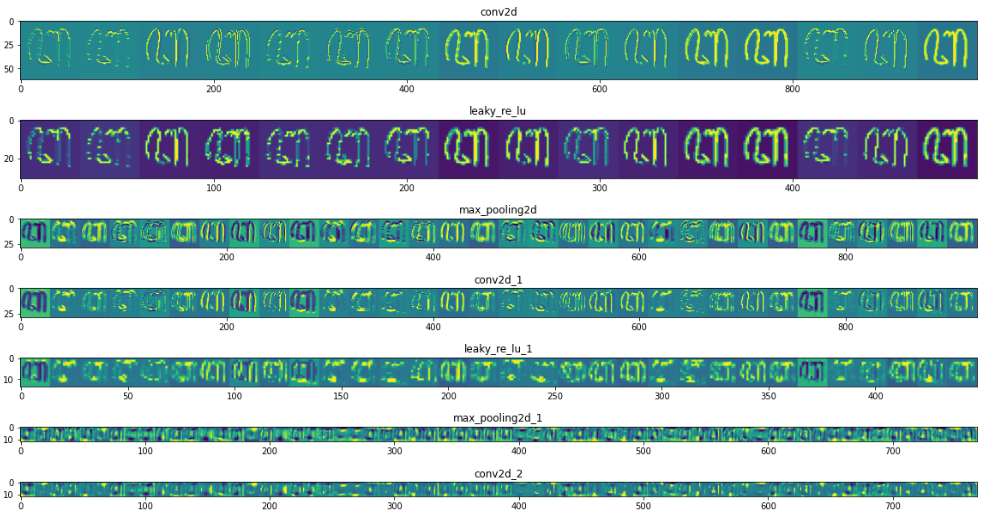
Non-trainable params: 0



On the accuracy & loss plot, the overfit problems seem overcomed. Both training accuracy and validation continue to rise until epoch 24. And validation loss is decreased overtime along with training loss.

With a smaller model, we can see the model performance is good on the test data. Accuracy is high, it is 96.9%. And the precision is 97.48%, recall: 96.9%.





From the activation layer plot, you can see the image of the results of convolution between layers is clearly visible, this indicates that the neurons are active, this is better than the previous results using relu activation.

1. **Documentations - how to replicate our works**

#### 

#### Duplicate the web app source code

* The web application source code is on [deploy\_aksara](https://github.com/biyanoscar/bangkit-dps4c/blob/master/deploy_aksara) folder
* To using this app on local machine, run app using this command to start Flask API

python main.py

The application will run on port 5000.

* Navigate to URL http:/127.0.0.1:5000

#### **Deploying web app to Google App Engine**

1. Create a Google Cloud account
2. On local computer, install the gcloud SDK/CLI

Install the App Engine extension  
gcloud components install app-engine-python

1. Set the project as your gcloud default with this command. Change MY-PROJECT-ID according to your respective PROJECT-ID  
   gcloud config set project MY-PROJECT-ID
2. Enable the Cloud Build API  
   gcloud services enable cloudbuild.googleapis.com
3. Initialize App Engine for our project  
   gcloud app create --project=MY-PROJECT-ID
4. Deploy to App Engine  
   gcloud app deploy
5. **Local (Indonesia) implementation of our project**

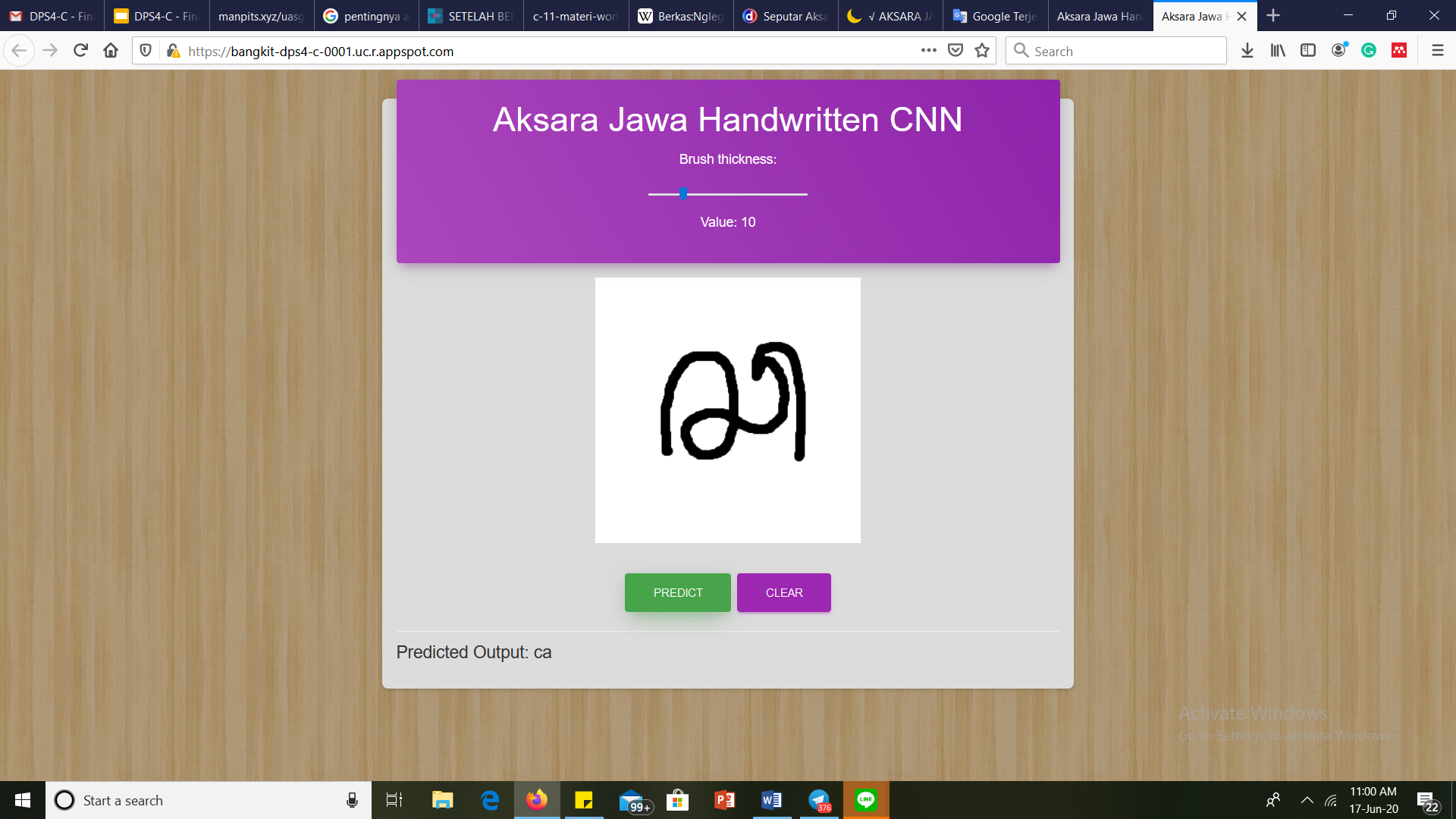
The idea of local implementation of our project is to make an educational application for learning Javanese script.Users can write the script pattern in the application, then it will display the results.

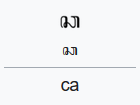
We create flask application that use our keras model. We deployed the model and application on Google App Engine.

You can access application on <https://bangkit-dps4-c-0001.uc.r.appspot.com/>

This application is a classification of Aksara Jawa based on the results of Javanese script pattern of users that have been provided on the application and will be searched for the pronunciation of the script.

Implementation of the application starts from the selection of brush thickness, and the user can create Javanese script whose pronunciation is known by pressing the predict button, and if you want to delete the image results can use the clear button





Implementation of the application by trying to write the script "ca" on the application and the application can execute well

1. **Paper Reference**

**Paper**

Xu, Bing; Wang, Naiyan; Chen, Tianqi; Li, Mu (2015, November 27). *Empirical Evaluation of Rectified Activations in Convolutional Network*

<https://arxiv.org/pdf/1505.00853.pdf>

Gu, Shanqing; Pednekar, Manisha; and Slater, Robert (2019) "*Improve Image Classification Using Data Augmentation and Neural Networks,*" SMU Data Science Review: Vol. 2 : No. 2 , Article 1.

<https://scholar.smu.edu/cgi/viewcontent.cgi?article=1091&context=datasciencereview>