

Image classification

COS30082 – Applied Machine Learning



March 5, 2025

Name: hong anh nguyen

STUDENT ID: 104359382

# Methodology

In this section, the experience will flow from basic to advanced. The advanced methodologies such as augmentation data, transfer learning, systematic testing system…

***Overview of data collector:***

* The data first split physically to 2 folders (train and test) with ratio 9:1, the train folder further split to train and validation with ratio 8:2.
* Then these data were loaded using following method:

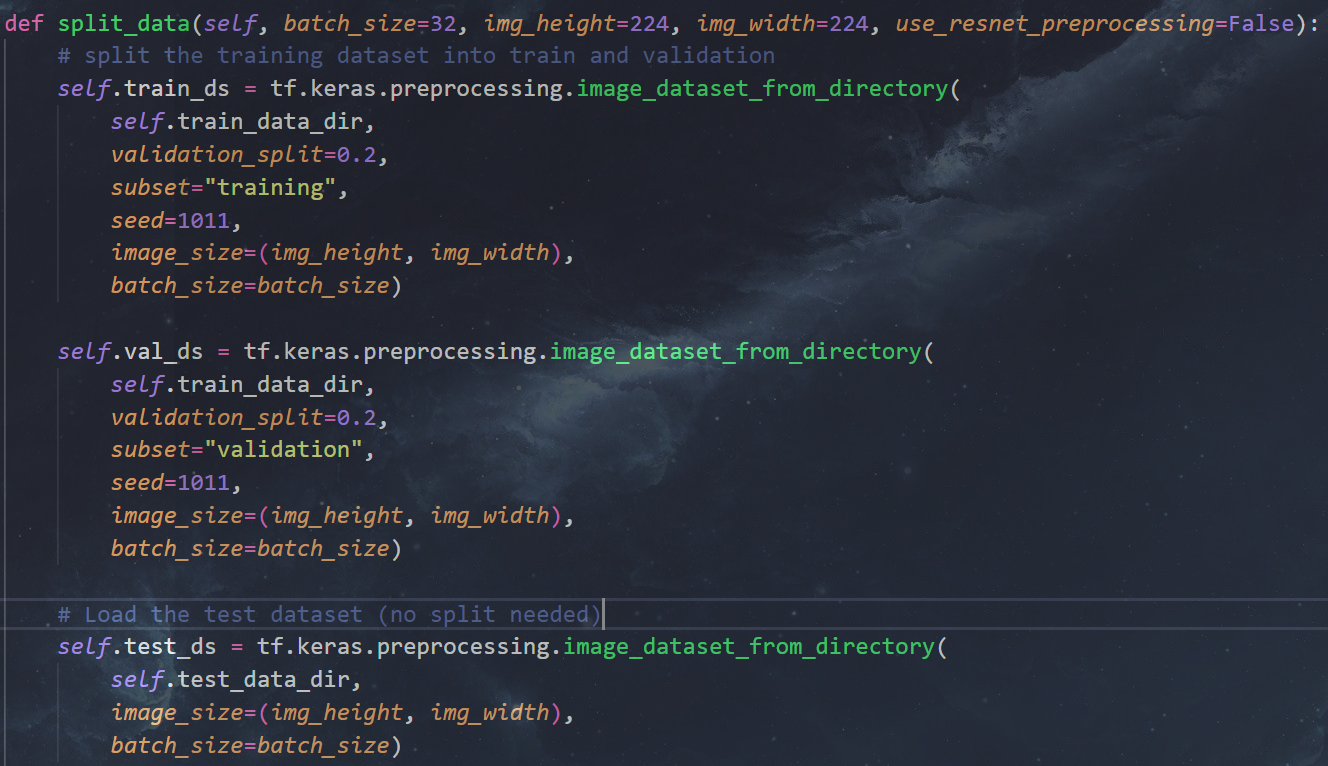


Figure . Loading and splitting Data

* Data augmentation setup:

A screen shot of a computer

AI-generated content may be incorrect.

Figure . Data augmentation

* Preprocessing data (using normal standard normalization or using preprocessing method provided by ResNet):

A screen shot of a computer program

AI-generated content may be incorrect.

Figure . Preprocessing Data

* Finally, configuration the autotune for faster loading:

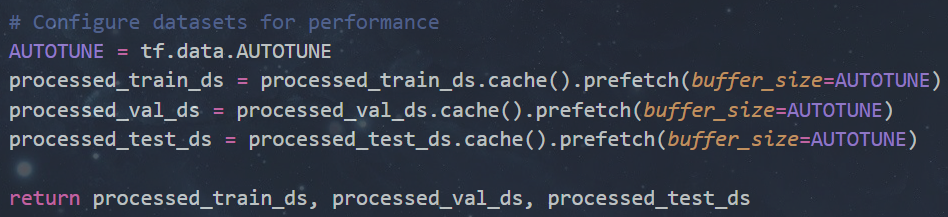


Figure . Autotune and return the datasets

***Overview of Training method:***

****

Figure . Callback constrains

* Early stopping with 10 epochs patient, Checkpoint for best model using validation accuracy and ReduceLROnPlateau learning rate scheduler were used. The reason for this scheduler is this is the most general strategy since if for certain epochs, no lower validation loss found then reduce the learning rate by a scaling factor. Since in this project, both pre-trained model and from scratch model were trained, using general scheduler will make it more convenient for the training process.
* A class weights strategy was implemented (since at first, I thought the dataset was imbalance but, it very balance 1000 images each class)

A computer screen with text

AI-generated content may be incorrect.

Figure . class weight calculates for weight training

* Optimizer using Adaldelta, loss using sparse categorical crossentropy and metric using accuracy. On this project, I trained model using local machine (Window Native) which only TensorFlow 2.10 and before support making some function like optimizer, metrics not up to date (Adam don’t have decay, and lack of some other methods which I planned to implement such as Label smoothing). After experimenting several optimizers such as Adam, SGD and Adadelta, the Adadelta showed the best performance with stable training.

A screen shot of a computer

AI-generated content may be incorrect.

Figure . Training hyper-params

1. Basic CNN experience

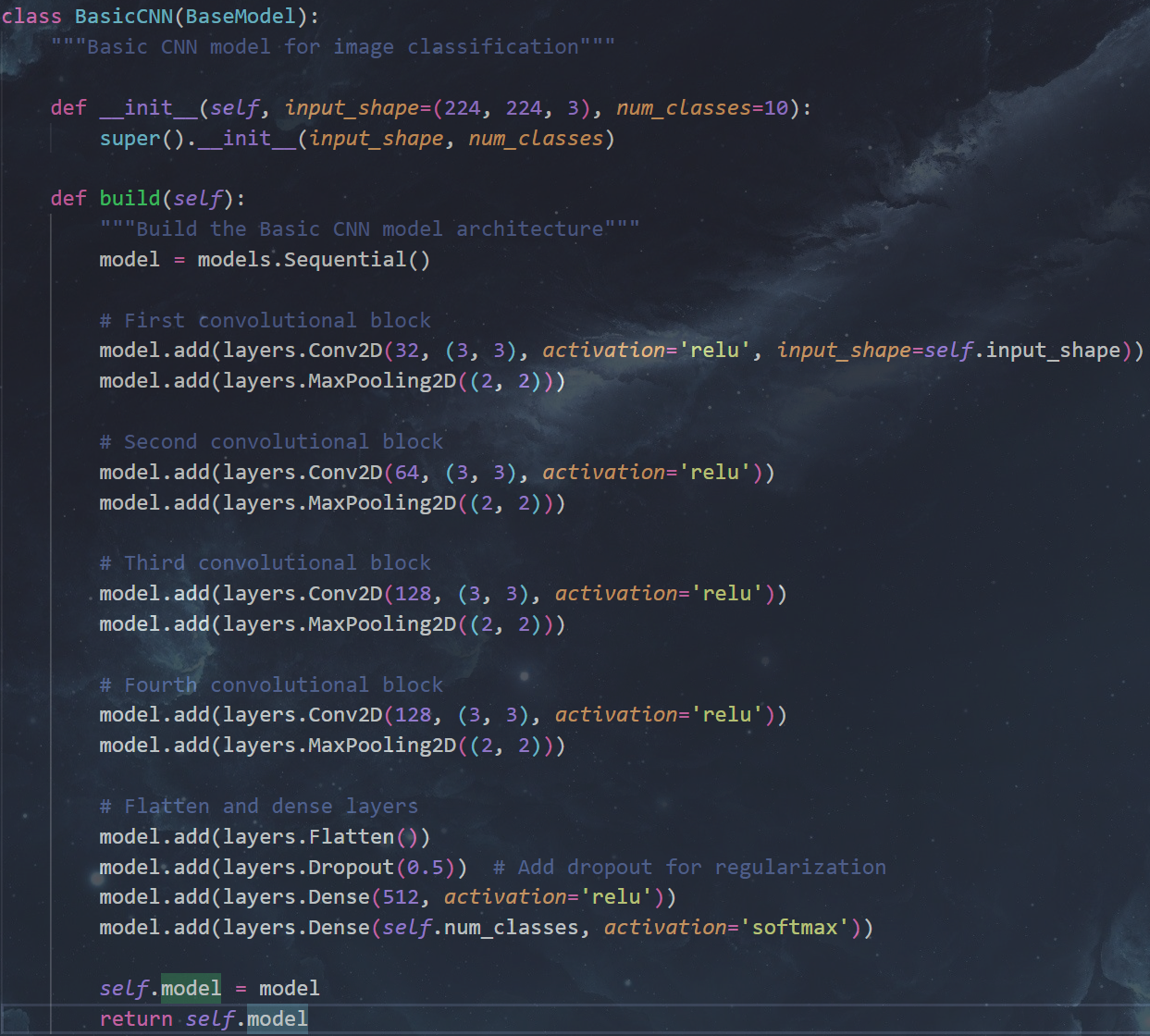


Figure . Basic CNN’s architecture

The basic CNN architecture was simply random initiated without reference to achieve some overview about the quality of the data (e.g. How is the diversity, quantity and quality).

# Results and Discussion

* Basic CNN:
* A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

  AI-generated content may be incorrect.
* Figure 9. Training and Evaluating History
* A screenshot of a graph

  AI-generated content may be incorrect.
* Figure 10. Testing results – Basic CNN
* Resnet 50 from scratch:

A graph of a line

AI-generated content may be incorrect.

Figure . Training and Evaluating History

A screenshot of a calendar

AI-generated content may be incorrect.

Figure . Testing result – ResNet50 from scratch

* Resnet 50 pre-trained, fine-tune with 20 layers unfreeze:

A graph of a graph of a graph

AI-generated content may be incorrect.

Figure . Training and Evaluating History

A screenshot of a graph

AI-generated content may be incorrect.

Figure . Testing result – Resnet50 fine-tuned