**HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FACULTY OF HIGH QUALITY TRAINING**

****

**FINAL ESSAY**

**MAJOR: ARTIFICIAL INTELLIGENCE**

**RESEARCH, DESIGN AND BUILD AI MODEL FOR**

**CLASSIFYING MOTORCYCLES USING**

**CONVOLUTIONAL NEURAL NETWORK**

**TRINH MINH TRI 20146288**

**Advisor: NGUYEN TRUONG THINH, Ph.D**

**Ho Chi Minh City, 24th of May 2023**

**TABLE OF CONTENTS**

**Chapter 1: Introduction 1**

* 1. Introduction 1
  2. Related researches 1
  3. Research goal 1
  4. Research method 1

**Chapter 2: Methodlogy 2**

2.1. Collecting data 2

2.2. CNN method description 2

2.3. CNN model architecture used to classify motorcycles 2

2.3.1. The Convolutional layer 3

2.3.2. The Pooling layers 3

2.3.3. The Fully Connected Layer 4

**Chapter 3: Model and algorithm 5**

3.1. Build the dataset 5

3.2. Train model 6

3.3. Accuracy of the model 7

3.4. Test the model 8

**Chapter 4: Results and discussion 10**

4.1. Results 10

4.2. Discussion 10

**Chapter 5: Conclusion 11**

**List of references 12**

**CHAPTER 1: INTRODUCTION**

* 1. **Introduction**

Nowadays, there are a lot of motorcycles on Vietnamese roads. Motorcycles had become a very popular type of transportation in Viet Nam in recent years. There are more and more motorcycle suppliers in Viet Nam. As a result, the need for a computer program to help the suppliers to manage the types of product they have and control the number of each type is very high. To solve this problem, the application of AI is very useful. I have trained an AI model to classify motorcycles according to their manufacturer, name, and color.

* 1. **Related researches**

Researches in the field of classifying objects, including vehicles, has been proceeded in many years, and there are a lot of different methods has been proposed. Using Convolutional Neural Network (CNN) to classify vehicles has become a popular and useful choice. Compare with traditional methods that classify objects according to their features, CNN has give higher accuracy. A research has compared CNN with K-nearest neighbor method (KNN) in classifying objects and the result has shown that CNN has given higher classifying accuracy than KNN. In conclusion, using CNN to classify vehicles, including motorcycles, has become an advanced and effect method.

* 1. **Research goal**

1. Build an AI model to classify motorcycles according to their manufacturer, name, and color with high accuracy.
2. Build a computer program which includes the model to do the task.
3. Evaluate the model and upgrade the model for later use.
   1. **Research method**

Research and analyze theory: Collect documents from sources of newspapers, magazines, books, from the internet related to the research content.

Experimental method: Conduct design, build, test operation and accuracy of the AI model, computer program, and complete the related design.

**CHAPTER 2: METHODLOGY**

**2.1. Collecting data**

I have collected a variety of photos about 17 types of motorcycles according to their manufacturer, name, and color to make the database. The request of the task is to classify them, so I use over 72 300x300 pixel images of each type of motorcycle. Before training the model, I do the data preprocessing steps to prepare for the training steps.

**2.2. CNN method description**

Classifying motorcycles using CNN is a problem in the fields of machine vision and artificial intelligence. It required building a CNN model to classify objects in photos, which are predefined categories.

To solve this problem, first of all, we need a big database which contains photos of different motorcycles. This database will be divided into two set: training set and test set. Training set will be used to train CNN model, otherwise, test set will be used to evaluate the accuracy of the model.

Then, a fit CNN model is designed and trained in the training set. This model will automatically learn how to distinguish the features of different motorcycles in the training images. After that, the model is used to classify motorcycles in the test set.

The result of the classifying motorcycles using CNN problem is evaluated according to the accuracy of the classification in the test set. A good CNN models high classifying accuracy in the test set, so that it has the ability to classify and detect objects in different images.

**2.3. CNN model architecture used to classify motorcycles**

The CNN architecture comprises three main layers: convolutional layers, pooling layers, and a fully connected (FC) layer. There can be multiple convolutional and pooling layers. The more layers in the network, the greater the complexity and (theoretically) the accuracy of the machine learning model. Each additional layer that processes the input data increases the model’s ability to recognize objects and patterns in the data.



*Fig 1: Example of CNN model architecture*

**2.3.1. The Convolutional layer**

Convolutional layers are the key building block of the network, where most of the computations are carried out. It works by applying a filter to the input data to identify features. This filter, known as a feature detector, checks the image input’s receptive fields for a given feature. This operation is referred to as convolution.

The filter is a two-dimensional array of weights that represents part of a 2-dimensional image. A filter is typically a 3×3 matrix, although there are other possible sizes. The filter is applied to a region within the input image and calculates a dot product between the pixels, which is fed to an output array. The filter then shifts and repeats the process until it has covered the whole image. The final output of all the filter processes is called the feature map.

The CNN typically applies the ReLU (Rectified Linear Unit) transformation to each feature map after every convolution to introduce nonlinearity to the ML model. A convolutional layer is typically followed by a pooling layer. Together, the convolutional and pooling layers make up a convolutional block.

Additional convolution blocks will follow the first block, creating a hierarchical structure with later layers learning from the earlier layers. For example, a CNN model might train to detect cars in images. Cars can be viewed as the sum of their parts, including the wheels, boot, and windscreen. Each feature of a car equates to a low-level pattern identified by the neural network, which then combines these parts to create a high-level pattern.

**2.3.2.** **The Pooling layers**

A pooling or downsampling layer reduces the dimensionality of the input. Like a convolutional operation, pooling operations use a filter to sweep the whole input image, but it doesn’t use weights. The filter instead uses an aggregation function to populate the output array based on the receptive field’s values.

There are two key types of pooling:

* Average pooling: The filter calculates the receptive field’s average value when it scans the input.
* Max pooling: The filter sends the pixel with the maximum value to populate the output array. This approach is more common than average pooling.

Pooling layers are important despite causing some information to be lost, because they help reduce the complexity and increase the efficiency of the CNN. It also reduces the risk of overfitting.

**2.3.3. The Fully Connected Layer**

The final layer of a CNN is a fully connected layer. The FC layer performs classification tasks using the features that the previous layers and filters extracted. Instead of ReLu functions, the FC layer typically uses a softmax function that classifies inputs more appropriately and produces a probability score between 0 and 1.

**CHAPTER 3: MODEL AND ALGORITHM**

**3.1. Build the dataset**

from os import listdir  
from numpy import asarray  
from numpy import save  
from keras.utils import load\_img, img\_to\_array  
folder = '/content/drive/MyDrive/AI/Cuoi\_ky\_AI/Vehicle/'  
photos, labels = list(), list()  
for file in listdir(folder):  
 output = 0.0  
 if file.startswith('Honda\_Vision\_Black '):  
 output = 1.0  
 if file.startswith('Honda\_Vision\_Red '):  
 output = 2.0  
 if file.startswith('Honda\_Vision\_White '):  
 output = 3.0  
 if file.startswith('Honda\_Wave\_Black '):  
 output = 4.0  
 if file.startswith('Honda\_Wave\_Red '):  
 output = 5.0  
 if file.startswith('Honda\_Wave\_White '):  
 output = 6.0  
 if file.startswith('Winner\_Black '):  
 output = 7.0  
 if file.startswith('Honda\_Winner\_Red '):  
 output = 8.0  
 if file.startswith('Exciter\_Black '):  
 output = 9.0  
 if file.startswith('Exciter\_Blue '):  
 output = 10.0  
 if file.startswith('Exciter\_Red '):  
 output = 11.0  
 if file.startswith('Janus\_Black '):  
 output = 12.0  
 if file.startswith('Yamaha\_Janus\_Red '):  
 output = 13.0  
 if file.startswith('Janus\_White '):  
 output = 14.0  
 if file.startswith('Sirius\_Black '):

output=15.0  
if file.startswith('Sirius\_Red '):  
 output = 16.0  
if file.startswith('Sirius\_White '):  
 output = 17.0  
img = load\_img(folder + file, target\_size = (300,300))  
photo = img\_to\_array(img)  
photos.append(photo)  
labels.append(output)

photos = asarray(photos)  
labels = asarray(labels)  
print(photos.shape,labels.shape)  
save('Vehicle\_photos\_train.npy', photos)  
save('Vehicle\_labels\_train.npy', labels)  
(1282, 300, 300, 3) (1282,)

**3.2. Train model**

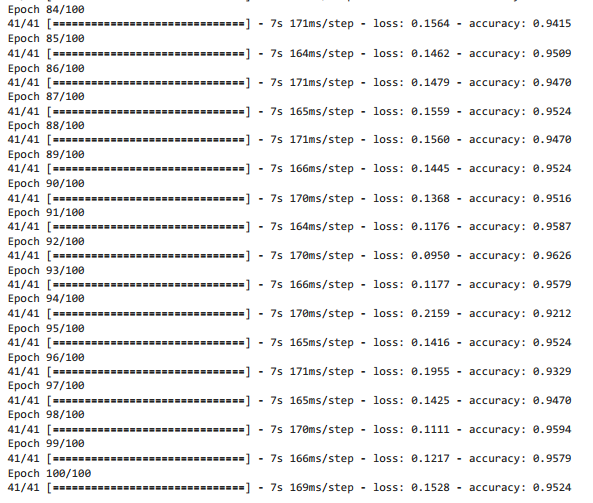
import numpy as np  
x\_train = np.load('/content/drive/MyDrive/AI/Cuoi\_ky\_AI/Vehicle\_photos\_train.npy')  
y\_train = np.load('/content/drive/MyDrive/AI/Cuoi\_ky\_AI/Vehicle\_labels\_train.npy')

x\_train = x\_train.astype('float32')/255  
from keras.utils import to\_categorical  
y\_train = to\_categorical(y\_train, 100)

from keras.models import Sequential  
from keras.layers import Dense,Dropout,Flatten,Conv2D,MaxPooling2D,Normalization,LeakyReLU  
from keras.optimizers import Adam  
# 32 lan tich chap  
model = Sequential()  
model.add(Conv2D(32,kernel\_size = (3,3),activation = 'relu',input\_shape=(300,300,3),padding='Same'))  
model.add(MaxPooling2D((2,2),padding='same'))  
model.add(Dropout(0.25))  
# 64 lan tich chap  
model.add(Conv2D(64,(3,3),activation ='relu',padding ='same'))  
model.add(MaxPooling2D((2,2),padding='same'))  
model.add(Dropout(0.25))  
# 128 lan tich chap  
model.add(Conv2D(128,(3,3),activation ='relu',padding ='same'))  
model.add(MaxPooling2D((2,2),padding='same'))  
model.add(Dropout(0.25))  
# 256 lan tich chap  
model.add(Conv2D(256,(3,3),activation ='relu',padding ='same'))  
model.add(MaxPooling2D((2,2),padding='same'))  
model.add(Dropout(0.25))  
# 512 lan tich chap  
model.add(Conv2D(512,(3,3),activation ='relu',padding ='same'))  
model.add(MaxPooling2D((2,2),padding='same'))  
model.add(Dropout(0.25))  
model.add(Flatten())  
model.add(Dense(64,activation = 'relu'))  
model.add(Dropout(0.25))  
model.add(Dense(100,activation='softmax'))  
from keras.losses import categorical\_crossentropy  
model.compile(loss = categorical\_crossentropy,optimizer = Adam(),metrics=['accuracy'])  
model.summary()  
train = model.fit(x\_train, y\_train,batch\_size=32, epochs = 100,verbose = 1)

model.save('Vehicle\_train.h5')

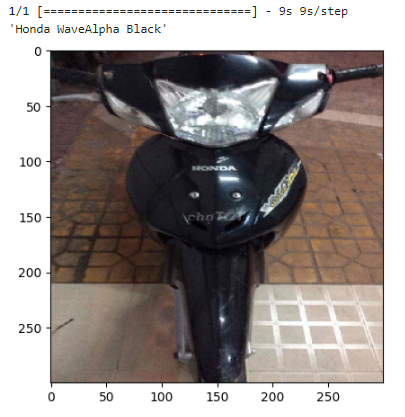
**3.3. Accuracy of the model**



*Fig 2: Accuracy of the model*

**3.4. Test the model**

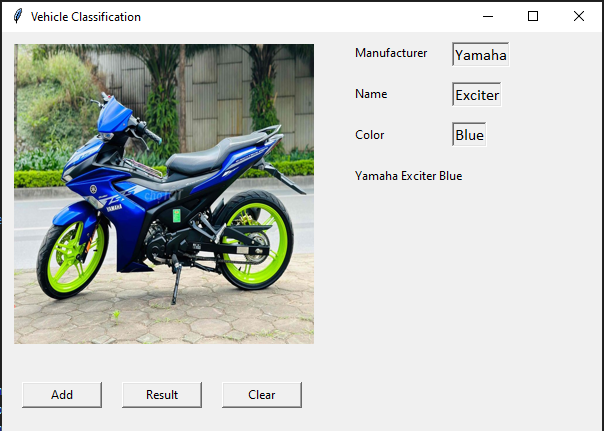
from keras.models import load\_model  
model1 = load\_model('/content/drive/MyDrive/AI/Cuoi\_ky\_AI/Vehicle\_train.h5')  
import matplotlib.pyplot as plt  
from keras.utils import load\_img  
from keras.utils.image\_utils import img\_to\_array  
import numpy as np  
vat = {1: 'Honda Vision Black', 2:'Honda Vision Red', 3:'Honda Vision White',  
4:'Honda WaveAlpha Black', 5:'Honda WaveAlpha Red', 6:'Honda WaveAlpha White',  
7:'Honda WinnerX Black', 8:'Honda WinnerX Red',  
9:'Yamaha Exciter Black', 10:'Yamaha Exciter Blue', 11:'Yamaha Exciter Red',  
12:'Yamaha Janus Black', 13:'Yamaha Yanus Red', 14:'Yamaha Janus White',  
15:'Yamaha Sirius Black', 16:'Yamaha Sirius Red', 17:'Yamaha Sirius White'}  
img = load\_img("/content/drive/MyDrive/AI/Cuoi\_ky\_AI/test/wave\_den (4).jpg",target\_size=(300, 300))  
plt.imshow(img)  
img = img\_to\_array(img)  
img = img.reshape(1,300,300,3)  
img = img.astype('float32')  
img =img/255  
result = np.argmax(model1.predict(img),axis=1)  
vat[result[0]]



*Fig 3: Result of model testing*

**CHAPTER 4: RESULTS AND DISCUSSION**

**4.1. Results**



*Fig 4: Result of computer program testing*

**4.2. Discussion**

Although model and computer program testing given high accuracy performance, they can be updated to work more accuracy and be friendlier to the user. Firstly, we can add more images of each types of motorcycles to the training set, so that, the accuracy of the model is higher and less misunderstanding. Secondly, we can add more images of other types of motorcycles, so that the model can classify more types of vehicles. Thirdly, we can redesign the computer program’s window so that it is more colorful and friendlier to the user.

**CHAPTER 5: CONCLUSION**

According to the train and test results of the CNN model, the accuracy on the test set is 95.24%, with the lost function is 0.1528. This shows that the model learns how to classify different type of motorcycles from the training data and has the ability to classify right type of motorcycles on new images.

From these results, I conclude that using CNN model to classify vehicles, especially motorcycles, which is an effective method and could be applies in a variety of applications in life. CNN model has the ability to research patterns of images data and classify motorcycles with high accuracy.

**LIST OF REFERENCES**

1. Datagen, “Convolutional Neural Network: Benefits, Types, and Applications”, 2023. [Online]. Available: <https://datagen.tech/guides/computer-vision/cnn-convolutional-neural-network/> [Accessed: 25/05/2023].

My Github’s QR code:

