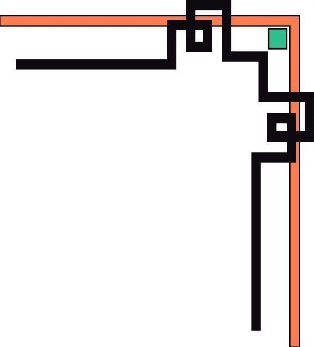
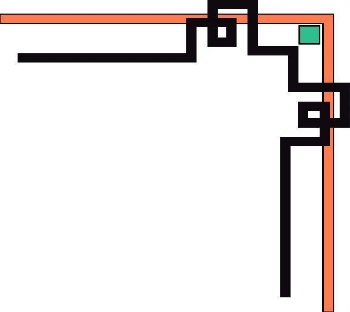
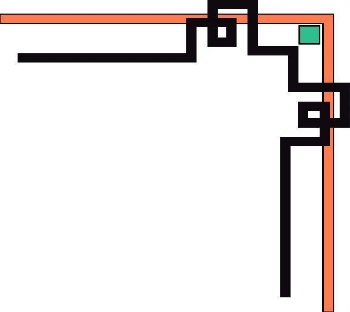
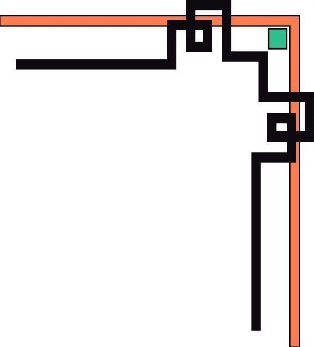
**HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION**



[**FACULTY OF MECHANICAL ENGINEERING**](http://fme.hcmute.edu.vn/?ArticleId=a69fffb8-3942-463b-9300-137c89a9213c)

**✡🕮🕮✡🕯**

****

**REPORT: ARTIFICIAL INTELLIGENCE**

**CLASSIFICATION AND GRADING SYSTEM FOR KOI FISH IN REAL TIME**

**LECTURER: Nguyen Truong Thinh**

**Student: Id**

**Tran Dinh Nam 19134079**

**Thu Duc City, December 2022**

**INSTRUCTOR'S COMMENTS**

**Project: Classification and grading system for koi fish in real time**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Student** | **Product completion rate** | **Phone** |
| **1** | **Tran Dinh Nam** | **100%** | **0366697867** |

**Intructor’s comments:**

**Ho Chi Minh, December 2022**

**(Grading teacher)**

## **Acknowledgement**

Learning must go hand in hand with practice, for students, the practice of theoretical lessons is extremely necessary, so the final project of this artificial intelligence course is to apply the theories that the teacher has taught. in class into practice. I would like to thank Mr. Nguyen Truong Thinh for creating conditions for me to research and complete this report. Especially, I would like to express my deep gratitude to the teacher who wholeheartedly imparted useful knowledge and guided us to carry out this thesis. I have tried to apply the theoretical knowledge learned in class in combination with the teacher's suggestions to complete this report. But due to limited knowledge and experience, my report still has many shortcomings. I hope to receive your reviews and suggestions. Once again, I would like to thank you for your help and concern during the preparation of this report.

Thank you very much, Mr. Nguyen Truong Thinh!

**INTRODUCTION 3**

**CHAPTER 1: OVERVIEW 5**

**1.1 Topic Introduction 6**

**1.2 Objectives of the study 6**

**1.3 Research scoped 6**

**CHAPTER 2: THEORETICAL BASIS 7**

**2.1 Convolutional neural networks 7**

**2.1.1 Introduction about CNN 7**

**2.1.2 Convolution Layer 8**

**2.1.3 Stride 9**

**2.1.4 Padding and Relu 10**

**2.1.5 Pooling Layer 10**

**2.2 Important Python Libraries Used 11**

**2.2.1 Matplotlib 11**

**2.2.2 Keras 12**

**2.2.3 Open CV 12**

**CHAPTER 3: CNN MODEL BUILDING 13**

**3.1 Model building 13**

**3.2 Model training 14**

**3.3 Show model quality 15**

**3.4 Save the model and verify when not using realtime 15**

**CHAPTER 4: RUN REALTIME FOR THE PROGRAM 16**

**CHAPTER 5: CONCLUSION AND DEVELOPMENT 17**

**5.1 Conclusion 17**

**5.2 Development 17**

**REFERENCE 18**

**INTRODUCTION**

Artificial intelligence is flourishing in an age where everything is geared towards intelligence. Currently, Artificial Intelligence systems in practice have been using artificial neural networks, which are a complex neural network that simulates the closest resemblance to a human neural network. people and animals. These networks can learn from and improve on their own experiences by changing connections in the same way that human and animal nerves work. When it comes to AI, we often think of something very far away, but in fact we are all exposed to it every day, it's Facebook's ad recommendation system, suggesting Movies we might want to see from other sites. Survey when we sign up for a Netflix account, google's voice search feature, or even google's spam filtering system and can see us in contact with AI every day.

From the reason and the growing importance of AI in a rapidly developing scientific and technological society today. The teachers in the Robotics industry have built the subject of Artificial Intelligence to help students in the industry catch up with the modern trend of the world. Through this subject, students have access to Artificial Intelligence algorithms and apply them to small homework exercises, so that students have a foundational knowledge to develop more in the process of going to work later. this. With the school's motto "LEARNING BY DOING", each student after completing the course will do a project to be able to apply the knowledge learned in 15 weeks into practice. After grasping the basic knowledge of the subject with the guidance from Mr. Nguyen Truong Thinh, I chose the topic " Classification and grading system for koi fish in real time". This is a topic that helps me to apply the knowledge I have learned and apply it in practice.

**CHAPTER 1: OVERVIEW**

**1.1 Topic Introduction**

The applications of image processing in daily life are the interesting fields that attract the most attention in the artificial intelligence industry with many highly applicable problems in reality. At the same time, with the strong development of deep learning algorithms, especially convolutional neural networks (CNN), have given outstanding results in testing problems. For example, in 2015, Kaiming He [8] proposed the ResNet network architecture and achieved a very good error rate of only 3.57%. In 2012, Alex and colleagues [7] proposed a model using CNN and won the ImageNet competition with an error rate of 15%. Koi fish are now being bought by many people, but distinguishing Koi fish is very difficult even for those who know it. From those reasons, I chose the topic "Classification and grading system for koi fish in real time".

**1.2 Objectives of the study**

- Human intelligence replicates.

- Create a system that predicts special human facial features that can be recognized well.

- After aggregating knowledge and developing additional unique set-up sequences.

**1.3 Research scope**

Create a Koi classification system using the available libraries of the Python programming language platform. This is a simple tool that makes it easy for users to create a system that recognizes facial features.

**CHAPTER 2: THEORETICAL BASIS**

**2.1 Convolutional neural networks**

**2.1.1 Introduction about CNN**

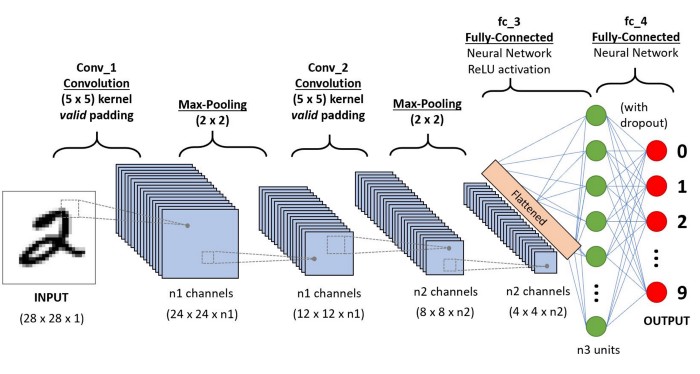


Figure 1: *CNN network structure*

The mathematical definition of convolution is a mathematical operation being applied on the two functions that give output in a form of a third function that shows how the shape of one function is being influenced, modified by the other function. The Convolutional neural networks (CNN) consists of various layers of artificial neurons. Artificial neurons, similar to that neuron cells that are being used by the human brain for passing various sensory input signals and other responses, are mathematical functions that are being used for calculating the sum of various inputs and giving output in the form of an activation value.

The behaviour of each CNN neuron is being defined by the value of its weights. When being fed with the values (of the pixel), the artificial neurons of a CNN recognizes various visual features and specifications. When we give an input image into a CNN, each of its inner layers generates various activation maps. Activation maps point out the relevant features of the given input image. Each of the CNN neurons generally takes input in the form of a group/patch of the pixel, multiplies their values(colours) by the value of its weights, adds them up, and input them through the respective activation function.

The first (or maybe the bottom) layer of the CNN usually recognizes the various features of the input image such as edges horizontally, vertically, and diagonally. The output of the first layer is being fed as an input of the next layer, which in turn will extract other complex features of the input image like corners and combinations of edges. The deeper one moves into the convolutional neural network, the more the layers start detecting various higher-level features such as objects, faces, etc

**2.1.2 Convolution Layer**

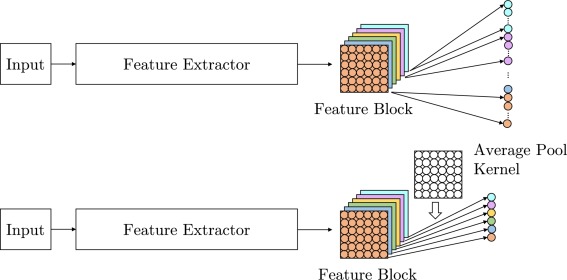


Figure 2: *Average pooling versus tensor flattening for conversion of spatial representation to feature embeddings.*

A convolutional layer is the main building block of a CNN. It contains a set of filters (or kernels), parameters of which are to be learned throughout the training. The size of the filters is usually smaller than the actual image. Each filter convolves with the image and creates an activation map. For convolution the filter slid across the height and width of the image and the dot product between every element of the filter and the input is calculated at every spatial position. Fig. 3.6 shows an example of the convolution process. The first entry of the activation map is calculated by convolving the filter with the portion marked blue in the input image. The activation map is generated by repeating this process for every element of the input image. The output volume of the convolutional layer is generated by stacking the activation maps of every filter along the depth dimension. Every component of the activation map can be thought to be the output of a neuron. Therefore each neuron is connected to a small local region in the input image, and the size of the area equals the size of the filter. All the neurons in an activation map also share parameters with each other. Due to the local connectivity of the convolutional layer, the network is forced to learn filters that have the maximum response to a local region of the input. The initial convolutional layers capture the low-level features (e.g., lines) of images, while the later layers extract the high-level features (e.g., shapes and specific objects)

**2.1.3 Stride**

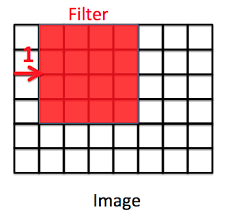
****

Figure 3: *Stride in CNN*

Stride is a component of convolutional neural networks, or neural networks tuned for the compression of images and video data. Stride is a parameter of the neural network's filter that modifies the amount of movement over the image or video. For example, if a neural network's stride is set to 1, the filter will move one pixel, or unit, at a time. The size of the filter affects the encoded output volume, so stride is often set to a whole integer, rather than a fraction or decimal.

**2.1.4 Padding and Relu**

There are two methods to help us deal when the kernel is not compatible with the original image, that is, inserting zeros into the shadow of the image or reducing the parts that do not match the kernel.

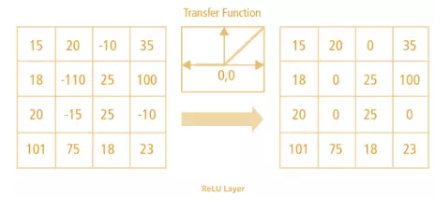


Figure 4: *Relu Layer*

Relu is a nonlinear function whose output is f(x) = max(0,x), this is a function that is used a lot when building CNNs. Relu presents nonlinearity in ConvNet as the data we normally use are linear values without sound. In addition, we also have the function tanh, sigmoid can also be used to replace the Relu function. However, with the advantage of having good performance, Relu is always used in building CNN models.

**2.1.5 Pooling Layer**

After the convolutional layer, it is a common practice to pass these values into the next layer which is known as the pooling layer. What pooling does is applying a special kind of filter to your output where the two most common form is max and average pooling. With max pooling, this is like applying a max filter on the image. This is also the same for the average pooling.

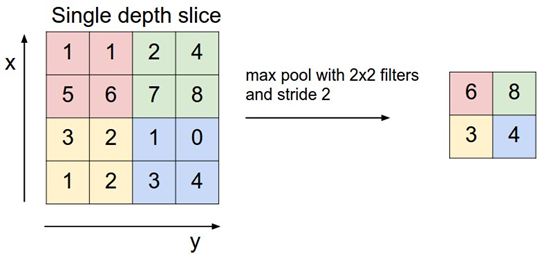
****

Figure 5: *Max pool with 2x2 filters and stride 2*

**2.2 Important Python Libraries Used**

**2.2.1 Matplotlib**

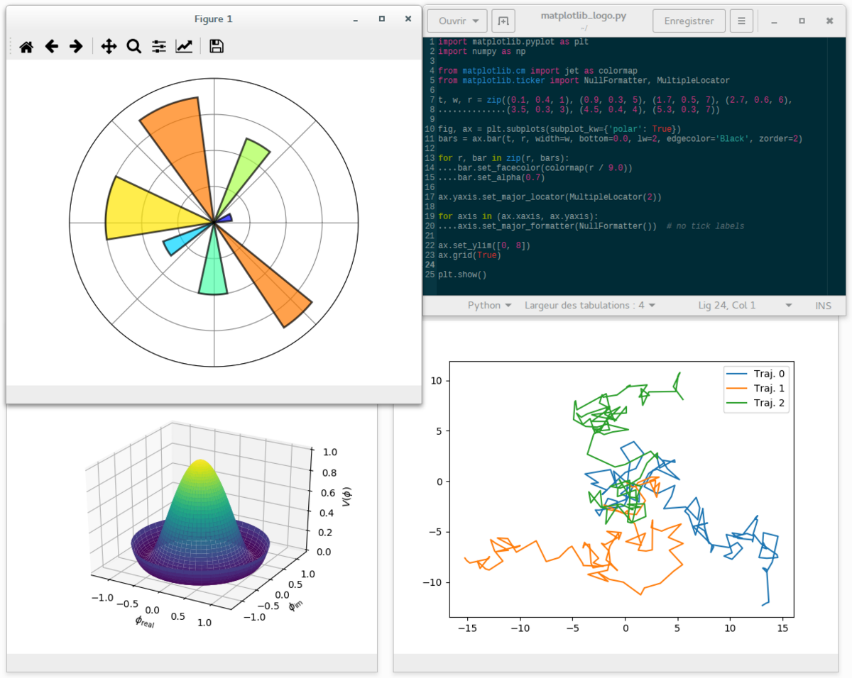


Figure 6: *Plotting using the matplotlib library*

In Python to visualize the data of the necessary statistical inferences, we can use Matplotlib, a very powerful and useful graphing library for those who work with Python and Numpy. Matplotib's most used module is Pyplot which provides an interface like MATLAB but instead it uses Python and it is open source. Matplotlib is suitable for presenting detailed information about processed data. Matplotlib brings your data to life with static, dynamic or interactive charts. When data is visualized graphically, programmers can easily present and express their programming ideas to customers (non-programmers).

**2.2.2 Keras**

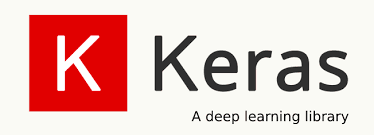


Figure 7: *Keras Library*

Keras is a neural network library written in python in 2015 by a google deep learning engineer. We can combine keras with deep learning libraries. (I often use a combination of keras and tensorflow).

**2.2.3 Open CV**

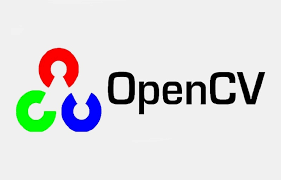
****

Figure 8: *Open CV library*

OpenCV is a leading open source library for image processing, machine vision and machine learning, and OpenCV has GPU acceleration features in real-time operation.

**CHAPTER 3: CNN MODEL BUILDING**

**3.1 Model building**

from keras.layers import Conv2D, MaxPooling2D

- Call some libraries needed to design neural networks

model = Sequential()

**-** Sequential model initialization.

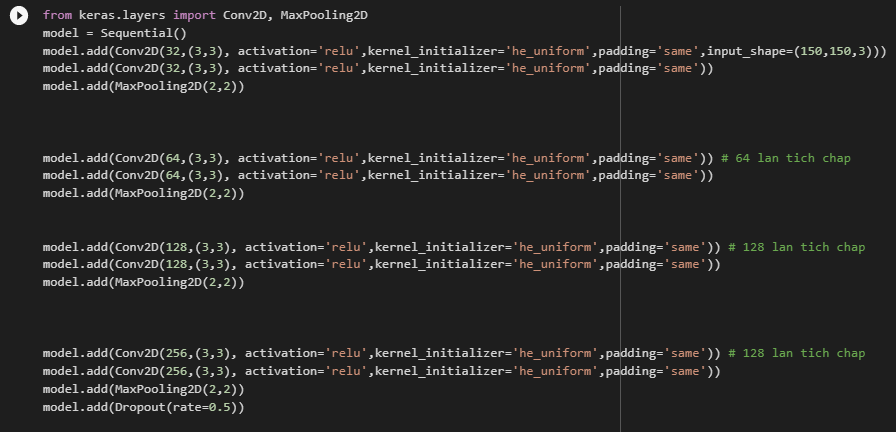


Figure 9: *Model building*

- The next three lines are creating CNN layer with 32 convolutions using Relu nonlinear function and MaxPooling2D pooling layer.

- The next three lines are to create CNN layer with 64 convolutions using Relu nonlinear function and MaxPooling2D pooling layer.

- The next three lines are to create CNN layer with 128 convolutions using Relu nonlinear function and MaxPooling2D pooling layer.

- The next three lines are to create CNN layer with 128 convolutions using Relu nonlinear function and MaxPooling2D pooling layer.

After using CNN to separate edges and now return to ANN type for classification.

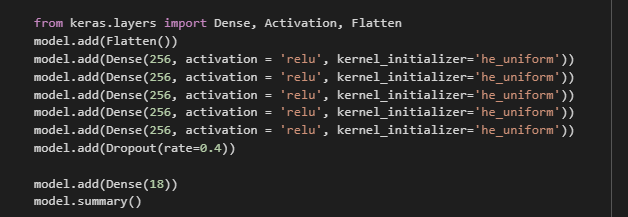


Figure 10: *Import Keras library*

- Use model.add(Flatten()) to flatten the layer to overlay the output convolution.

- The next three lines are to create a new hidden layer with the output spatial dimension of 128 and set relu as the activation function.

- The last layer with 2 output layers is Male or Female.

- The last line is to show the entire content after designing the neural network.

**3.2 Model training**

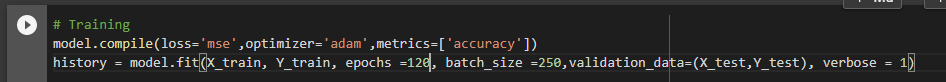
****

Figure 11: *Training model*

**3.3 Show model quality**

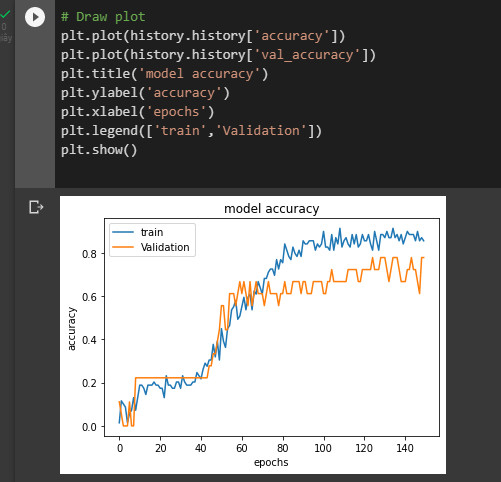


Figure 12: *Code that visualizes model quality graphically*

**3.4 Save the model and verify when not using realtime**

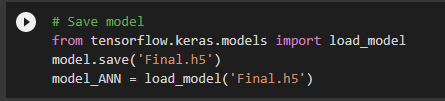
****

Figure 13*: Save Model with name Final.h5*

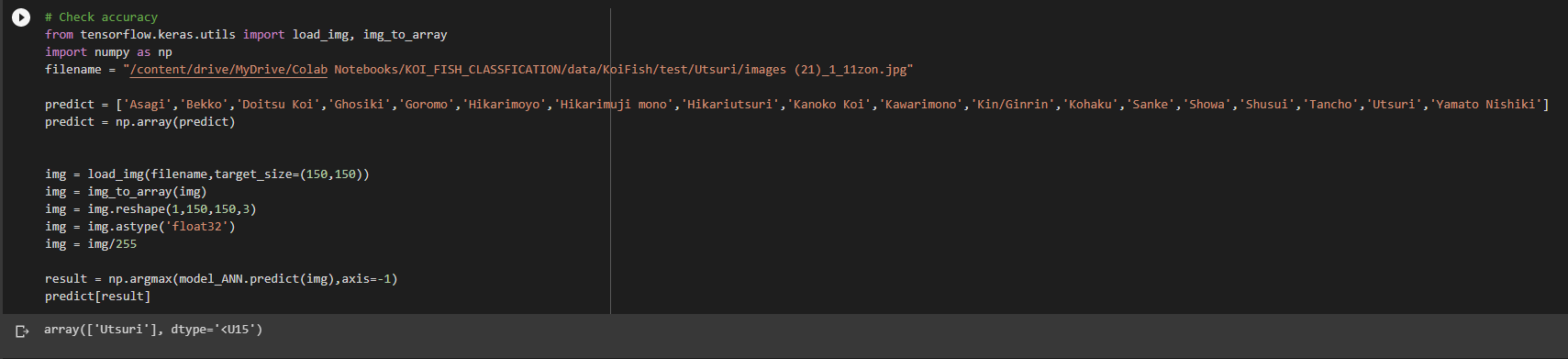
****

Figure 14: *Check accurary not real time*

**CHAPTER 4: RUN REALTIME FOR THE PROGRAM**

After completing the model training process and checking the accuracy of the model, the next step is to save the h5 file extension and save it to the realtime code to run it in real time to evaluate the quality of the model.

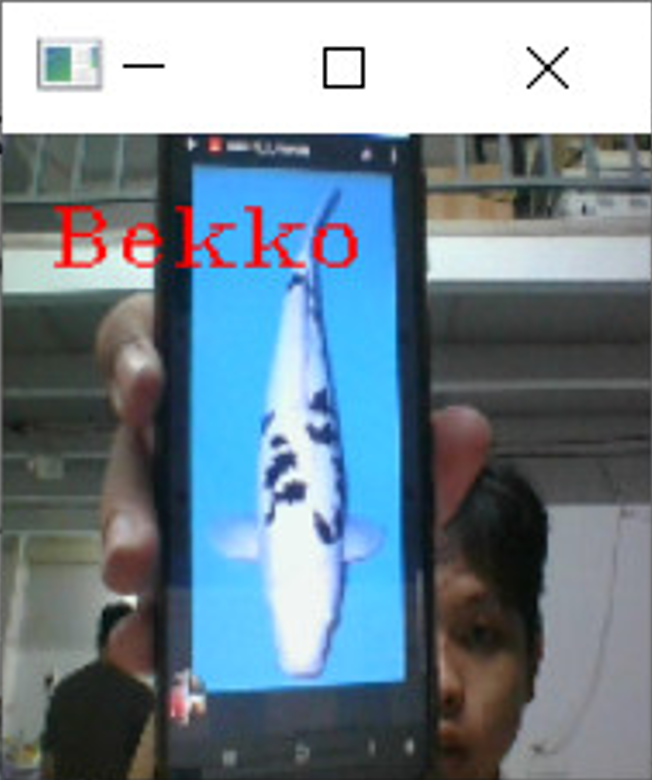


Figure 15: *Result run realtime*

**CHAPTER 5: CONCLUSION AND DEVELOPMENT**

**5.1 Conclusion**

- The model has implemented the proposed contents including:

+ Get an overview of the topic.

+ Programmatically build a model that predicts.

+ Use realtime to identify features.

+ Use the knowledge learned in class to apply to the project at the end of the semester.

- Advantages of the implementation project:

+ Gender and emotions predict well.

+ The results show the efforts of the implementer.

Disadvantages of the implementation project:

- The ability to predict koi is not very accurate.

- Easy to get noisy when encountering unfavorable scene in real time.

**5.2 Development**

- Build interface

- In the near future, I will develop a project so that I can put it outside the market to sell.

- Use and improve pre-train to increase the quality of training training model.

**REFERENCE**

[1] Deep Learning for Medical Image Analysis

### [2] Giới thiệu tổng quan về keras. Retrieved 11 August 2016

### <https://trituenhantao.github.io/2020/08/21/keras-la-gi-gioi-thieu-ve-keras/>

### [3] Ứng dụng Convolutional Neural Network trong bài toán phân loại ảnh. Retrieved 28 November 2017

### <https://viblo.asia/p/ung-dung-convolutional-neural-network-trong-bai-toan-phan-loai-anh-4dbZNg8ylYM>

### [4] [Deep Learning] Tìm hiểu về mạng tích chập (CNN). Retrieved 10 December 2020

### <https://viblo.asia/p/deep-learning-tim-hieu-ve-mang-tich-chap-cnn-maGK73bOKj2>

### [5] Stride (Machine Learning)

### <https://deepai.org/machine-learning-glossary-and-terms/stride>

### [6] Convolutional Layer

### <https://viblo.asia/p/ung-dung-convolutional-neural-network-trong-bai-toan-phan-loai-anh-4dbZNg8ylYM>

### [7] Basic Introduction to Convolutional Neural Network in Deep Learning. Retrieved 1 March 2020

### <https://www.analyticsvidhya.com/blog/2022/03/basic-introduction-to-convolutional-neural-network-in-deep-learning/>