University for the Creative Arts

BERLIN SCHOOL OF BUSINESS & INNOVATION

Essay / Assignment Title: Image classification project (set exercise)

Programme title:Data Analytics

Name:Safaa Kamaleldin Izzeldin Abdallah

Year:october/2022

CONTENTS

Statement of compliance with academic ethics and the avoidance of plagiarism

I honestly declare that this dissertation is entirely my own work and none of its part has been copied from printed or electronic sources, translated from foreign sources and reproduced from essays of other researchers or students. Wherever I have been based on ideas or other people texts I clearly declare it through the good use of references following academic ethics.

(In the case that is proved that part of the essay does not constitute an original work, but a copy of an already published essay or from another source, the student will be expelled permanently from the postgraduate program).

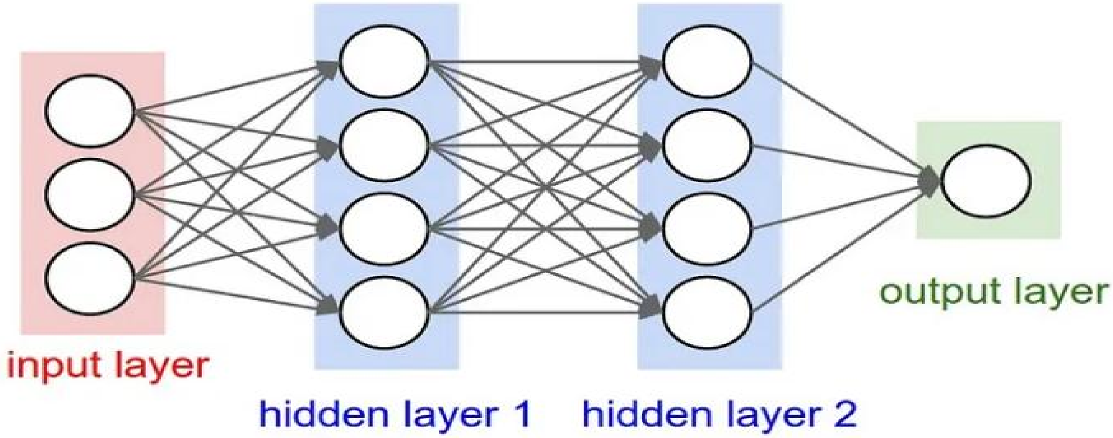
Name and Surname (Capital letters):

....SAFAA KAMALELDIN IZZELDIN ABDALLAH ....

Date: 15/April/2024

INTRODUCTION

Neural network in our word :

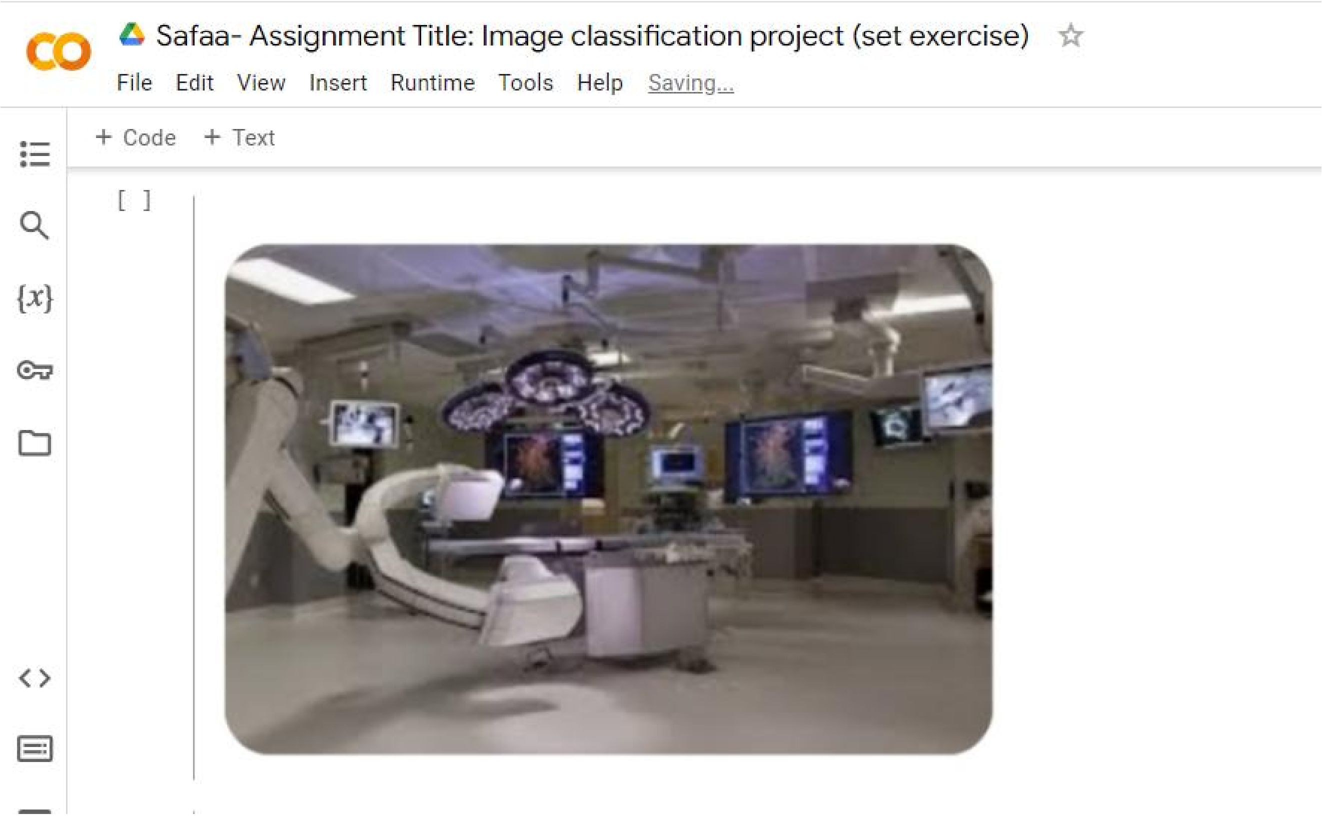


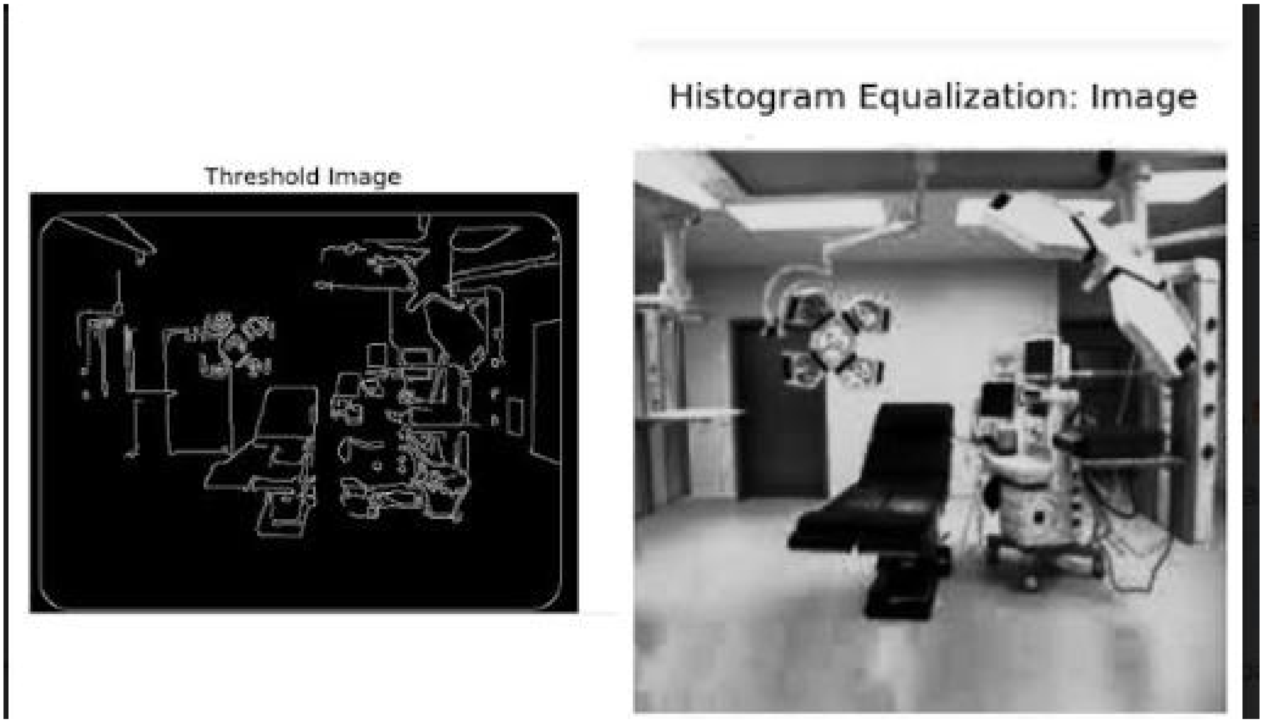
In the current era and amid all this development, it is not strange that technology has spread in all fields, and here in this work we will present to you the entry of technology and development into the world of medicine, medical institutions and health facilities, and this makes the matter more interesting because health is an indicator of the safety of society and the development of health facilities indicates this. The development and civilization of this society.

In this work, we have a robot inside an operating room in a hospital. We have worked on providing it with programming to detect and identify medical devices inside this room, and this is exactly similar to the human mind, which is a neural network taken from the ability of the Creator to man, the creations of the human mind in interpreting and perceiving images. And the landmarks around it and the ability to quickly recognize in response to actions, and this is exactly what NLP is trying to apply in the machine, and it is great for us that we are part of this interesting field of learning and work and very exciting and surprising in the continuous development of the field of machine learning and NLP.

Paper concept :

I prepared the program in a simple way and trained it on part of the equipment inside the operating room, such as the surgical table, the operating light, and the operating room as well. There are many other devices, but it is important to apply it to a sample of the work and review the results to avoid errors and know the behavior of the machine towards different programs and what is useful in This is the case of a robot, and then discussing the development and learning of the machine to make the most of it in all fields, not just health fields. I hope that this work finds the amount of excitement that surrounded us as we implemented it.


Safaa- Assignment Title: Image classification project (set exercise)
File
Edit
View
Insert
Runtime
Tools
Help
Saving...
+ Code
+ Text
[ ]


Threshold Image
Histogram Equalization: Image

CHAPTER ONE

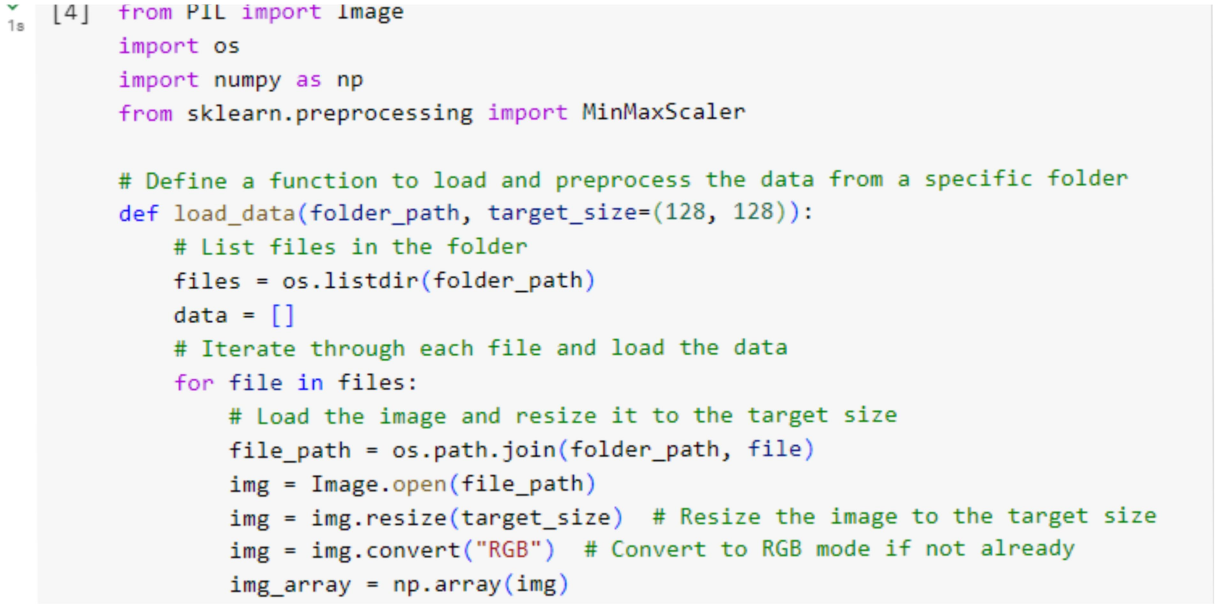
Data Preparation :

To implement this model we used google colab as the work environment, and then I prepared our own dataset RGB that we collected from different sources 251 images (operation room , operation table , surgery light) three category so configured to implement this task

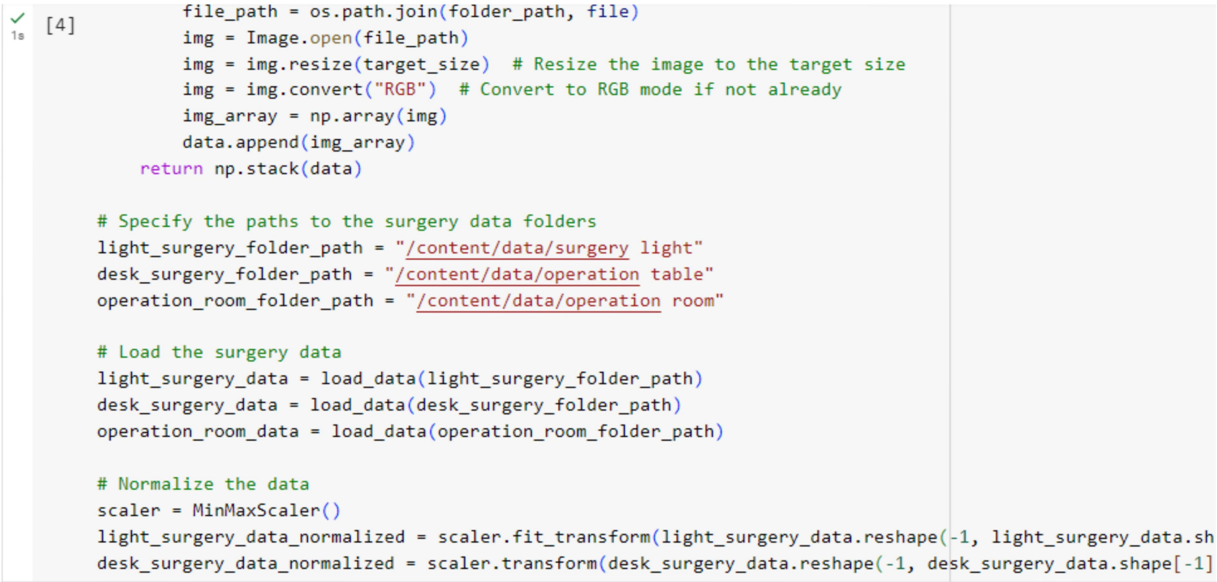
In this code Normalization , we imported the libraries required to work and built the model in the form required for the task, and reference was made to training on Tensorflow-Keras... We also defined all the parameters and devices and determined the image sizes to make them all of one size and shape and take them for training and then improve randomly to help the model perform well. Better and deal with the current images in training in a better way for better training results.... Flow _from _directory is to deal with specific directories and apply all the processors and transformations mentioned on them... and also define the CNN model, the classification neural network model of the sequence using the sequence programming interface and applications. The model consists of Several layers. This is to improve the performance of the model after training and increase accuracy. After the pelleting layers and pooling layers, then fully connected layers, the resulting layer has softmax activation units as an indication of the distribution of probabilities over the three classes. As for assembling the model here, we used Adam and his cross-classification loss function suitable for multi-class classification, such as This increases the accuracy measure. This code trains the fit model over and over again. The training and verification data generators determine the number of training steps for each training session and the number of training sessions in addition to the number of verification steps for each training session. Finally, this trained model is evaluated on this data for testing and printing accuracy. Testing: In general, we designed this model, which is based on class recognition and detection, image classification, and CNN model evaluation

Model Implement :

The model that I prepared is a neural network model for classifying images using the colors RGB, red, green, blue.... After preparing the training data and loading the images from the directories, applying optimization techniques to them, for example, changing the size to one fixed size 128*128 and normalizing it to a range of 1.0 and the structure of the model. Which consists of several layers. First, the spherification layers. These layers are used to extract features from the input images and use filters for this, known as kernels. Then, the maximum pooling layers. These layers reduce the dimensions of the feature maps that are collected by the spherification layers. Then the flat layer flattens this layer of spherification into a 1D vector. Then The full hard layers perform this classification based on the extracted features and then follow the structure of the model. This layer is alternating pelleting and pooling layers followed by the full hard layers. As for the pooling, it is done using the optimizer Adam, which here performs an effective optimization algorithm for training the neural network and uses the intersection of the classification regression as a loss function to solve multiple problems. We were I mentioned earlier fit to train the model, pass it through batches of data, and update the weight of the model to reduce the loss



```
[4] from PIL import Image
    import os
    import numpy as np
    from sklearn.preprocessing import MinMaxScaler

    # Define a function to load and preprocess the data from a specific folder
    def load_data(folder_path, target_size=(128, 128)):
        # List files in the folder
        files = os.listdir(folder_path)
        data = []
        # Iterate through each file and load the data
        for file in files:
            # Load the image and resize it to the target size
            file_path = os.path.join(folder_path, file)
            img = Image.open(file_path)
            img = img.resize(target_size)  # Resize the image to the target size
            img = img.convert("RGB")  # Convert to RGB mode if not already
            img_array = np.array(img)
```

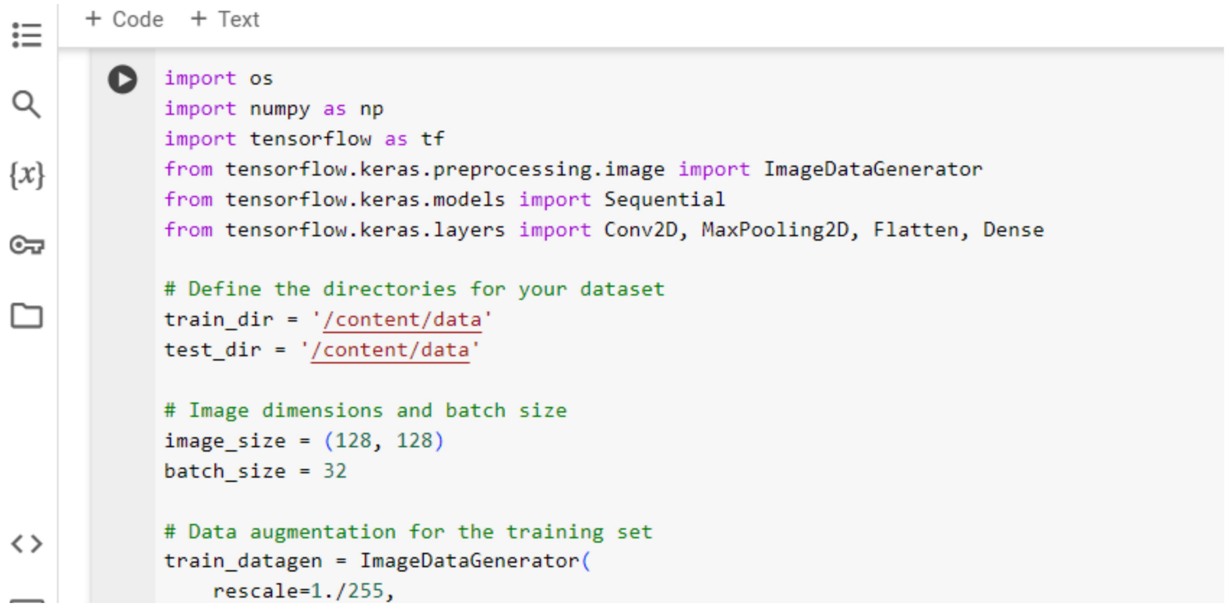


```
[4]         file_path = os.path.join(folder_path, file)
            img = Image.open(file_path)
            img = img.resize(target_size)  # Resize the image to the target size
            img = img.convert("RGB")  # Convert to RGB mode if not already
            img_array = np.array(img)
            data.append(img_array)
        return np.stack(data)

    # Specify the paths to the surgery data folders
    light_surgery_folder_path = "/content/data/surgery light"
    desk_surgery_folder_path = "/content/data/operation table"
    operation_room_folder_path = "/content/data/operation room"

    # Load the surgery data
    light_surgery_data = load_data(light_surgery_folder_path)
    desk_surgery_data = load_data(desk_surgery_folder_path)
    operation_room_data = load_data(operation_room_folder_path)

    # Normalize the data
    scaler = MinMaxScaler()
    light_surgery_data_normalized = scaler.fit_transform(light_surgery_data.reshape(-1, light_surgery_data.sh
    desk_surgery_data_normalized = scaler.transform(desk_surgery_data.reshape(-1, desk_surgery_data.shape[-1]
```

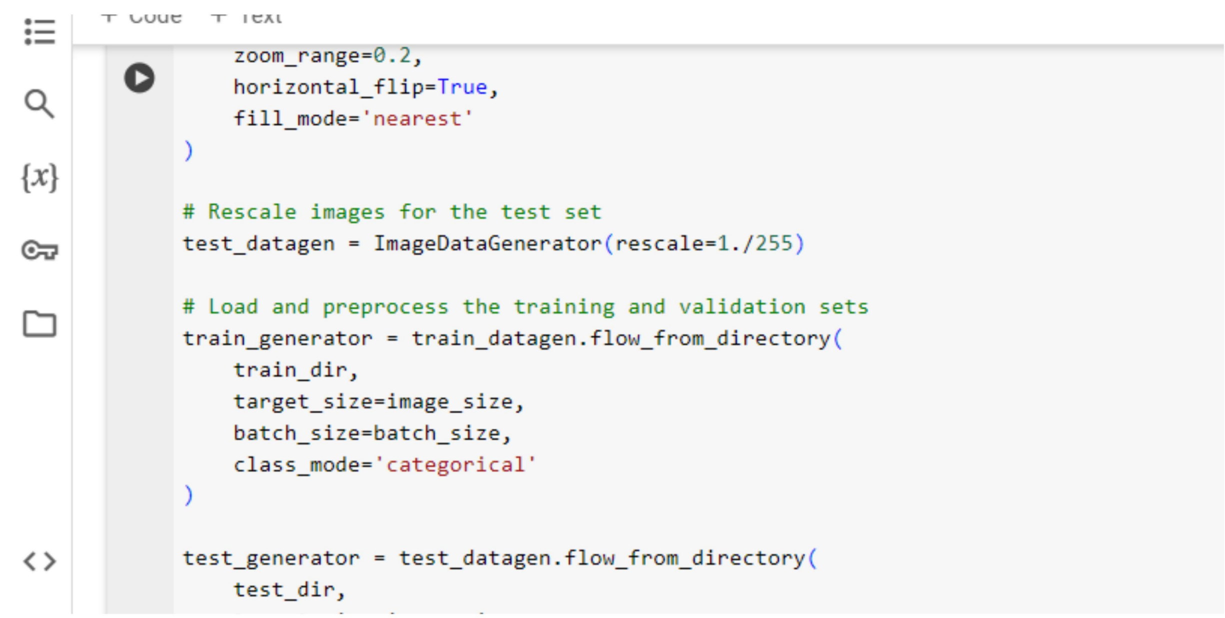
CHAPTER TWO

Model Training & Evaluation :

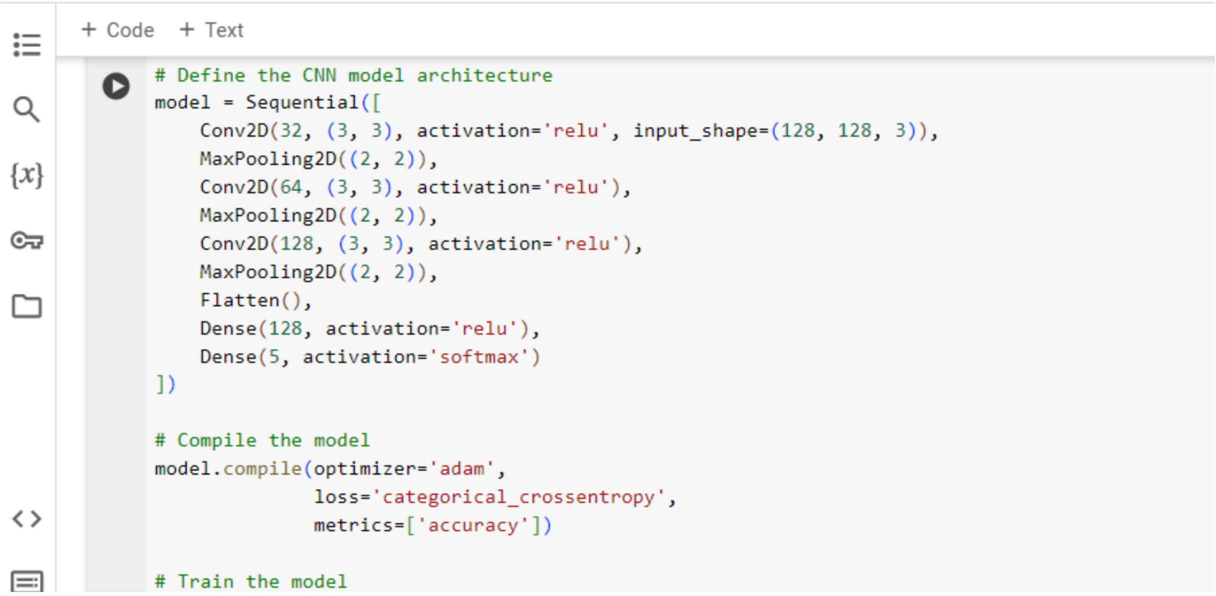
For training using Tensorflow-Keras, we called the necessary libraries to carry out the task of building and training the neural network, in addition to Numpy. After that, we defined the directories and parameters train_dir and test_dir, determined the path of the directories that contain the training and optional data sets, and the size of the images by specifying the dimensions to which the images will be scaled. batch_size specifies the number The images that are processed in each batch during training, train_datagen and test_datagen are from the imageDataGenerator class, which performs and improves the preparation of the data. Data optimization also includes operations such as rotation, moving, reflection and zooming, which help in increasing the diversity of the training eyes in the selection group. To print pixel values between 0.1 and The images are scaled and re-converted into decimal point compensation matrices and divided by 255 to normalize them. Definition of the CNN model using SEQUENTIAL to define the structure. The model consists of CON2VD followed by MaxPooling2D layers to extract features and images and reduce them. The last layer follows the rounding that controls Flatten D to convert the two-dimensional feature maps into a two-dimensional vector, which In turn, it is fed to the full solid layers (Dense). The last resulting layer contains three units due to the classification task. Softmax is used to extract the probability for each class and group the model using Adam, the classification loss function, the regression intercept, the accuracy measure, and the number of training cycles is 10 cycles.



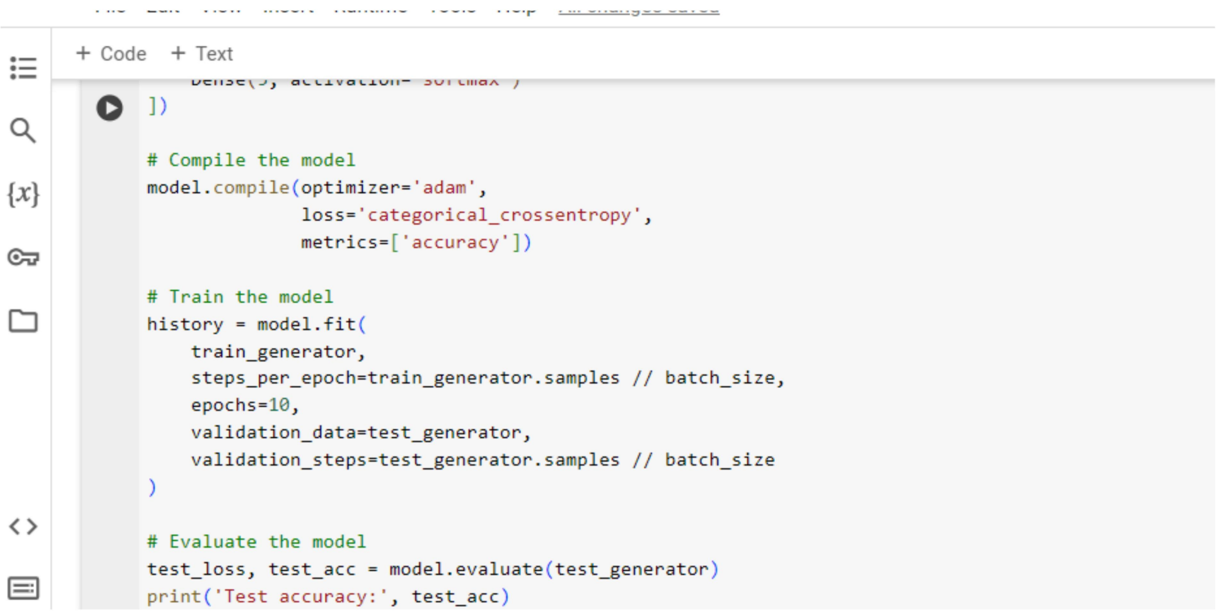
```
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

# Define the directories for your dataset
train_dir = '/content/data'
test_dir = '/content/data'

# Image dimensions and batch size
image_size = (128, 128)
batch_size = 32

# Data augmentation for the training set
train_datagen = ImageDataGenerator(
    rescale=1./255,
```



```
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

# Rescale images for the test set
test_datagen = ImageDataGenerator(rescale=1./255)

# Load and preprocess the training and validation sets
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical'
)

test_generator = test_datagen.flow_from_directory(
    test_dir,
```



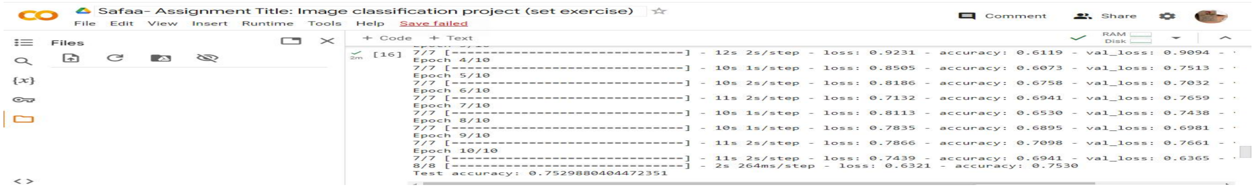
```python
# Define the CNN model architecture
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(128, 128, 3)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(5, activation='softmax')
])

# Compile the model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Train the model
```



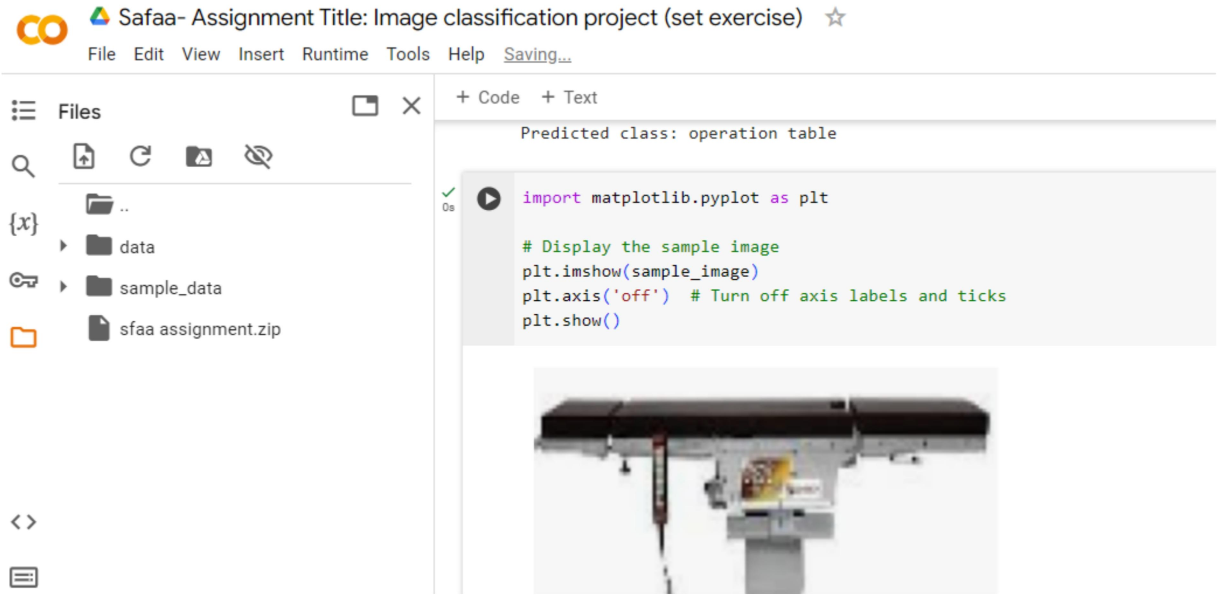
```python
    Dense(5, activation='softmax')
])

# Compile the model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Train the model
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // batch_size,
    epochs=10,
    validation_data=test_generator,
    validation_steps=test_generator.samples // batch_size
)

# Evaluate the model
test_loss, test_acc = model.evaluate(test_generator)
print('Test accuracy:', test_acc)
```

The accuracy :

The model after taring process shewing goog accuracy 75% that after we use Relu activation function and all the unique factors we use to create this model ..

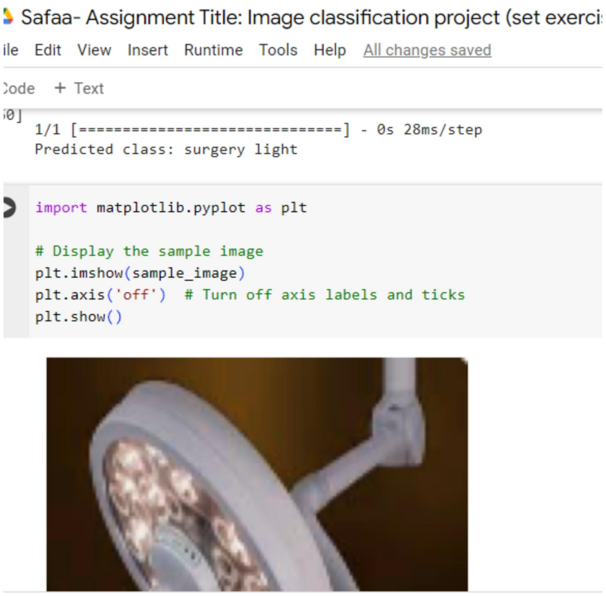


```
[16] 7/7 [==============================] - 12s 2s/step - loss: 0.9231 - accuracy: 0.6119 - val_loss: 0.9094 -
     Epoch 4/10
     7/7 [==============================] - 10s 1s/step - loss: 0.8505 - accuracy: 0.6073 - val_loss: 0.7513 -
     Epoch 5/10
     7/7 [==============================] - 10s 2s/step - loss: 0.8186 - accuracy: 0.6758 - val_loss: 0.7032 -
     Epoch 6/10
     7/7 [==============================] - 11s 2s/step - loss: 0.7132 - accuracy: 0.6941 - val_loss: 0.7659 -
     Epoch 7/10
     7/7 [==============================] - 10s 1s/step - loss: 0.8113 - accuracy: 0.6530 - val_loss: 0.7438 -
     Epoch 8/10
     7/7 [==============================] - 10s 1s/step - loss: 0.7835 - accuracy: 0.6895 - val_loss: 0.6981 -
     Epoch 9/10
     7/7 [==============================] - 11s 2s/step - loss: 0.7866 - accuracy: 0.7098 - val_loss: 0.7661 -
     Epoch 10/10
     7/7 [==============================] - 11s 2s/step - loss: 0.7439 - accuracy: 0.6941 - val_loss: 0.6365 -
     8/8 [==============================] - 2s 264ms/step - loss: 0.6321 - accuracy: 0.7530
     Test accuracy: 0.7529880404472351
```


Safaa- Assignment Title: Image classification project (set exercise)
File Edit View Insert Runtime Tools Help Saving...
Files
..
data
sample_data
sfaa assignment.zip
+ Code + Text
0s

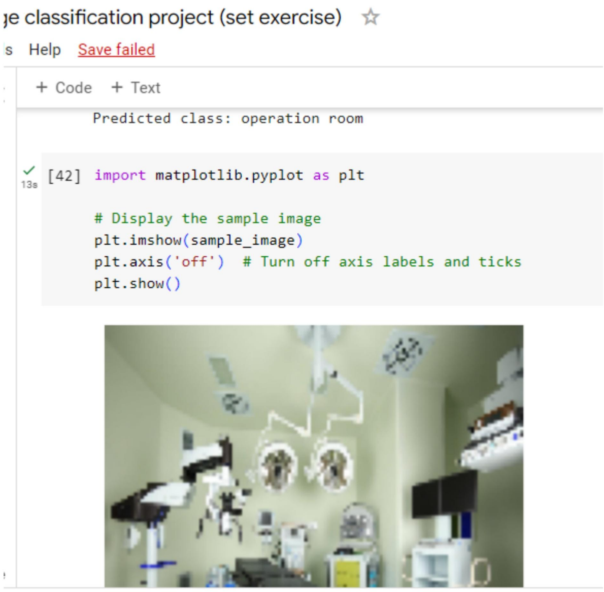
```
Predicted class: operation table
```

```
import matplotlib.pyplot as plt

# Display the sample image
plt.imshow(sample_image)
plt.axis('off')  # Turn off axis labels and ticks
plt.show()
```


Safaa- Assignment Title: Image classification project (set exerci
ile Edit View Insert Runtime Tools Help All changes saved
Code + Text

```
50]
 1/1 [==============================] - 0s 28ms/step
 Predicted class: surgery light
```

```
import matplotlib.pyplot as plt

# Display the sample image
plt.imshow(sample_image)
plt.axis('off')  # Turn off axis labels and ticks
plt.show()
```



```
Predicted class: operation room
```

```
[42] import matplotlib.pyplot as plt

     # Display the sample image
     plt.imshow(sample_image)
     plt.axis('off')  # Turn off axis labels and ticks
     plt.show()
```

Practical Application :

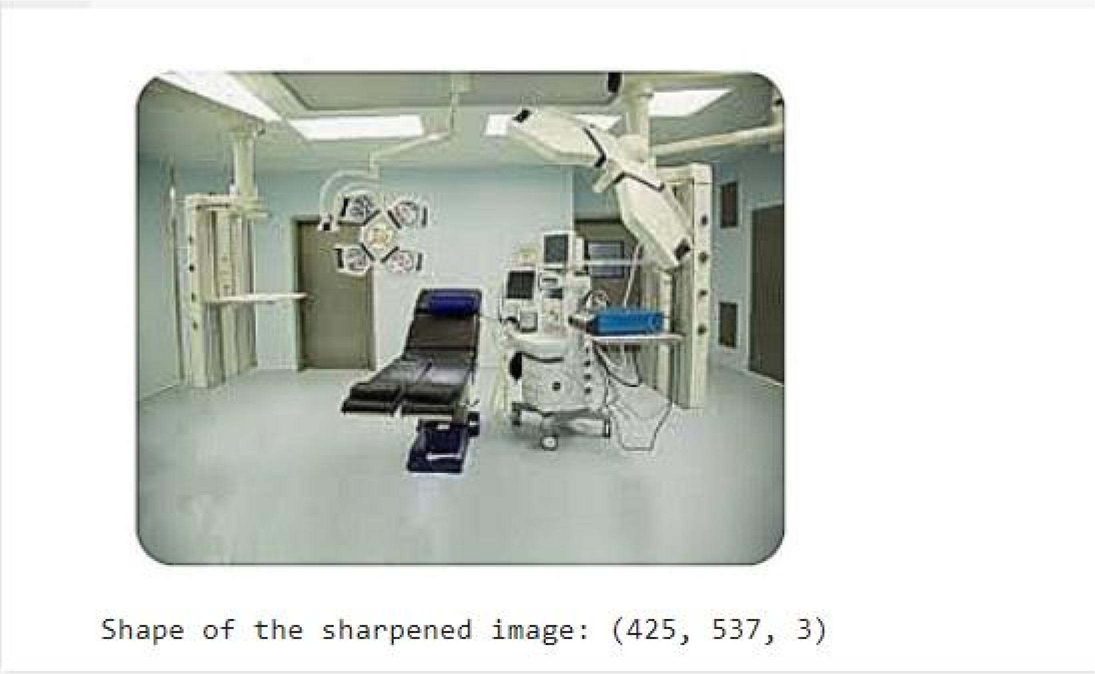
After application to the implemented model, it demonstrated an appropriate performance ability with the size of the data it was tested on and the experimental conditions, and the accuracy was 72%, a rather high percentage.

However, when applying some codes to add a high capacity to the model and noticing the differences, **canny edge detection** was very ideal in clarification and sorting. Elements in the image and detecting edges is an ideal option for communicating a signal to the robot. Edge detection is considered ideal in the case of my experience with the robot.

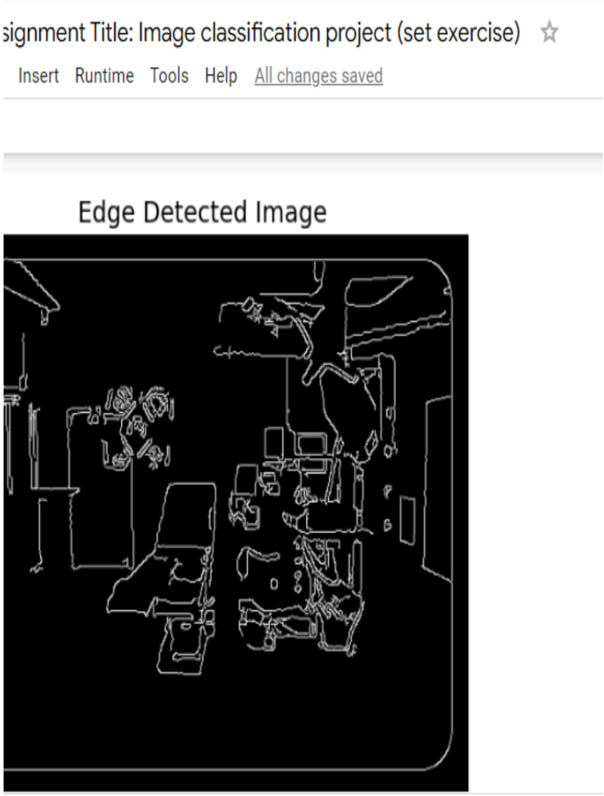
Histogram Equalization is to increase and raise the color equality in the image. It can be useful in some cases for this robot experiment, where it is possible to distinguish between elements more clearly. In neural programming of robots.

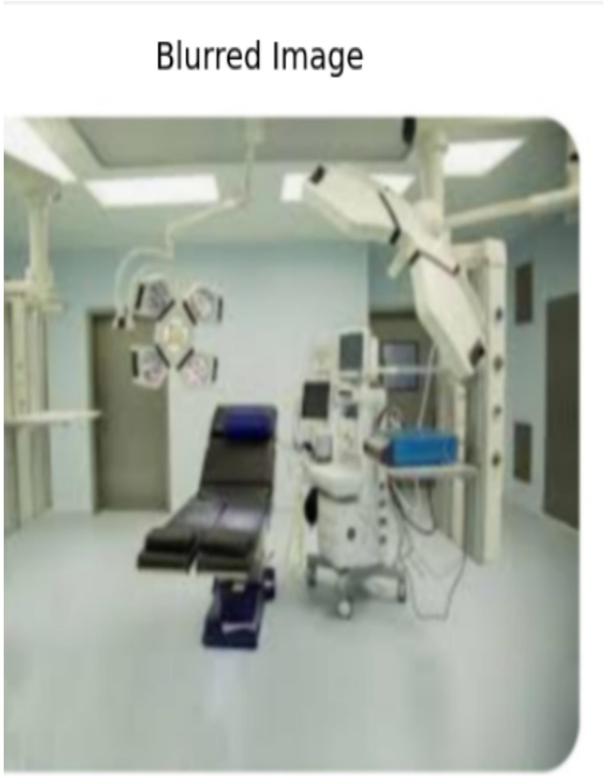
Gray cod This application in the case of robots inside the operating room is not very useful because the tasks are specific and not large, and because in cases where the details in the image are not clear regarding the colors it is useful.

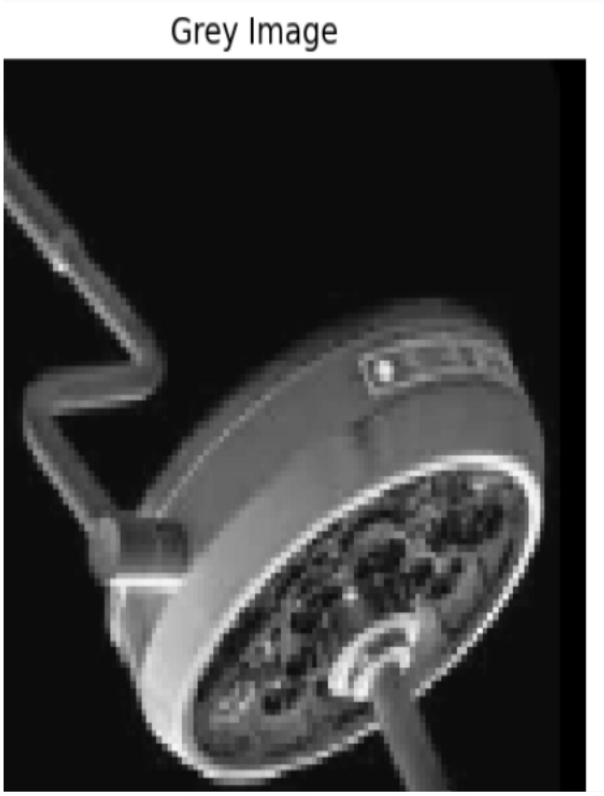
Blurred image It is possible to indicate The code is useful for reducing noise, but it is also not useful for the robot in the current experiment.

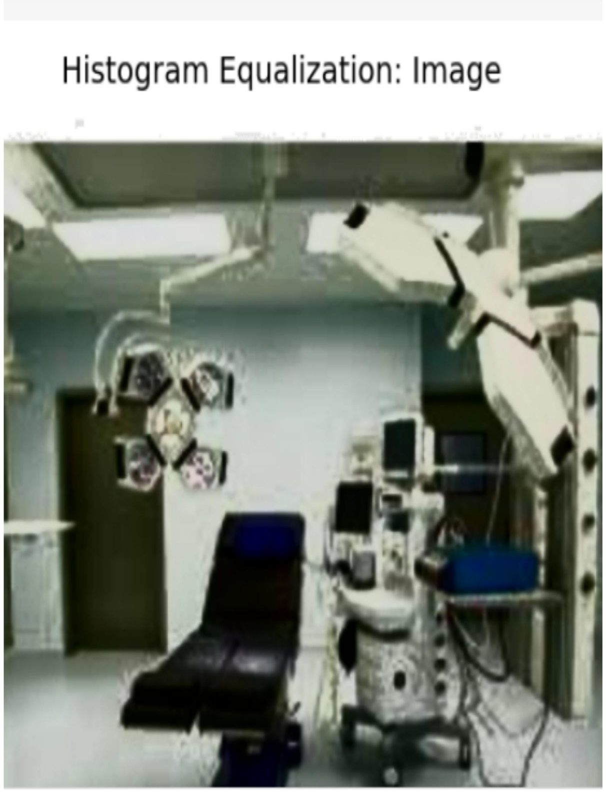


```
Shape of the sharpened image: (425, 537, 3)
```


signment Title: Image classification project (set exercise) ☆
Insert Runtime Tools Help All changes saved
Edge Detected Image


Blurred Image


Grey Image


Histogram Equalization: Image

CONCLUDING REMARKS

As for the program implemented for the robot to detect the operating room in the hospital and medical devices, the work on it was of great benefit to me in terms of implementation, and I hope that after participating it will also be useful....

In this program implemented by a neural network, which simulates the human being in behavior, neural interpretations, and cell connectivity. And signals, and forming reactions, and translating these signals into actions for communication....

Here we implemented a code whose goal is to program the robot to recognize objects inside the hospital room, and after preparing the collected data that I collected from individual sources, after initialization, the simple code was implemented, the results of which were impressive. It is very positive. It is an input layer containing 32 neutron, as well as two hidden layers in the middle for processing. The importance of these two layers is known, and the use of the activation function Relu, which is one of the most effective functions in image classification, has very positive results. 64 neutron for each layer in the sum of 128, and this is extremely effective in this code. The image classification is ideal, and the output layer was for the three classified categories, which are operation table, surgery light, operation room, and we use also softmax activation function to display the output after processing and executing the code.... After that, we obtained an accuracy that is considered appropriate for the size of the data, 72% after Training and implementing predictions on training to find out whether the robot is able to recognize objects or not. Experiments applied to the three trained categories proved that the robot gives correct results despite the lack of collected data, but its ability was high in reading and recognizing devices.....

After that, we conducted a number of applications. I looked at the images to get the maximum benefit and know their effectiveness, and the edge detected image, in my opinion, was the best because it was able to detect separate elements without distinguishing colors, but it is useful in detecting elements, and also histogram equalization is useful for another purpose, which is the sharpness of colors, and this is useful in distinguishing other elements. It has clear features and gray also has places that make it important as a program that serves the purpose of processing images, specifically in detecting resolution. We also applied the Blurred Image program, which removes noise from images and makes them flat, and is also useful for the purposes of detecting images.

Future Direction :

The accuracy and ability of the machine learning model can be improved and used in image and classification techniques, in addition to the ability to control the setting, description, selection of goals and their implementation accurately via machine learning neural networks. This robot can be more effective in the medical sector by controlling and programming it for more tasks and providing unparalleled benefit. It is certain that It provides a more expensive and safer service for health sector workers and patients as well

Reference:

Model link:

Safaa.A(2024) [https://colab.research.google.com/drive/1zSS2bGwvghH0hxxpDFm_vthPi9zls-rj?usp=sharing]

Data link :

Sfaa.A(2024) [https://drive.google.com/file/d/1Z_3fzB6EDw6qciLqLpEaebrn43XO-juz/view?usp=sharing]

Image:

BMC.Software[https://images.app.goo.gl/LunS2z1afWhiHLQj6]