

Team id	PNT2022TMID51236
Project Name	REAL TIME COMMUNICATION SYSTEM FOR SPECIALLY AIDED PEOPLES
Date and sprint no	3-11-2022 AND 3

Test the model

1) Import packages and load the model

The screenshot shows a Jupyter Notebook titled 'Untitled5' running on a local host. The code in the notebook performs the following steps:

- Imports `load_model` from `keras.models`, `numpy` as `np`, and `cv2`.
- Loads the model from a file named `as1png1.h5`.
- Creates three input images (`img1`, `img2`, `img3`) as `np.array` objects.
- Creates two kernels: `kernel_horizontal` and `kernel_vertical`, both as `np.array` objects.
- Prints the kernels and their intended uses for detecting horizontal and vertical edges.

```

Out[17]: <keras.callbacks.History at 0x1f743a90280>

In [18]: model.save('as1png1.h5')

In [3]: from keras.models import load_model
import numpy as np
import cv2

In [4]: model=load_model('as1png1.h5')

In [5]: img1 = np.array([np.array([200, 200]), np.array([200, 200])])
img2 = np.array([np.array([200, 200]), np.array([0, 0])])
img3 = np.array([np.array([200, 0]), np.array([200, 0])])

kernel_horizontal = np.array([np.array([2, 2]), np.array([-2, -2])])
print(kernel_horizontal, 'is a kernel for detecting horizontal edges')

kernel_vertical = np.array([np.array([2, -2]), np.array([2, -2])])
print(kernel_vertical, 'is a kernel for detecting vertical edges')

[[ 2  2]
 [-2 -2]] is a kernel for detecting horizontal edges
[[ 2  2]
 [ 2 -2]] is a kernel for detecting vertical edges

```

2) Pre-process and predict the image

Home Page - Select or create a x imageprocessing - Jupyter Note Untitled5 - Jupyter Notebook x +

localhost:8888/notebooks/Untitled5.ipynb

jupyter Untitled5 Last Checkpoint: 8 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)


In [18]:

```
from tensorflow.keras.models import load_model
import numpy as np
import cv2
from tensorflow.keras.preprocessing import image
```

In [19]:

```
img=image.load_img('C:\\Users\\ELCOT\\Downloads\\Dataset\\training_set\\A\\1.png',target_size=(400,500))
img
```

Out[19]:



28°C Mostly clear ENG 18:41

Home Page - Select or create a x imageprocessing - Jupyter Note Untitled5 - Jupyter Notebook x +

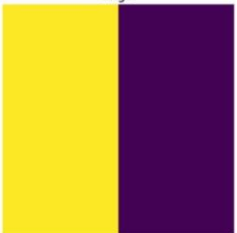
localhost:8888/notebooks/Untitled5.ipynb

jupyter Untitled5 Last Checkpoint: 8 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

kernel_vertical))

img3



Horizontal edge confidence score: 0
Vertical edge confidence score: 800

In [15]:

```
print("Len x-train : ", len(xtrain))
print("Len x-test : ", len(xtest))
```

Len x-train : 158
Len x-test : 23

28°C Mostly clear ENG 18:41

Home Page - Select or create a x imageprocessing - Jupyter Noteb x Untitled5 - Jupyter Notebook x +

localhost:8888/notebooks/Untitled5.ipynb

jupyter Untitled5 Last Checkpoint: 9 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

In [26]:

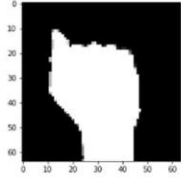
```
from skimage.transform import resize
def detect(frame):
    img=resize(frame,(64,64,1))
    img=np.expand_dims(img,axis=0)
    if(np.max(img)>1):
        prediction=model.predict(img)
        print(prediction)
        prediction=model.predict_classes(img)
        print(prediction)
```

In [27]:

```
arr= image.img_to_array(img)
import cv2
import matplotlib.pyplot as plt
```

In [30]:

```
frame=cv2.imread('C:\\Users\\ELCOT\\Downloads\\Dataset\\training_set\\A\\10.png')
data=detect(frame)
plt.imshow(frame)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



28°C Mostly clear ENG 18:53