

Ech - II

Deep learning

• Aim :

Implementation of deep learning algorithms

• Code :

```
import tensorflow as tf  
from tensorflow.keras.preprocessing.image  
import ImageDataGenerator  
from tensorflow.keras.applications import VGG16  
from tensorflow.keras.applications import layers  
models  
base_model = VGG16(weights='imagenet', include_top=  
=False, input_shape=(224, 224, 3))  
train_dir = '/kaggle/input/cats-and-dogs/train'  
validation_dir = '/kaggle/input/cats-and-dogs/val'  
train_datagen = ImageDataGenerator(  
    rescale=1. / 255,  
    rotation_range=40,  
    width_shift_range=0.2,  
    height_shift_range=0.2,  
    shear_range=0.2,  
    zoom_range=0.2,  
    horizontal_flip=True,  
    fill_mode='nearest'  
)
```

```
test_datagen = ImageDataGenerator(rescale=1. / 255)  
train_generator = train_datagen.flow_from_directory(  
    train_dir,  
    target_size=(224, 224),  
    batch_size=20,  
    class_mode='binary'
```

```
validation_generator = test_datagen.flow_from_directory
```

```
validation_dir,
```

```
target_size=(224, 224),
```

```
batch_size=20,
```

```
class_mode='binary'
```

```
)
```

```
model = models.Sequential()
```

```
model.add(base_model)
```

```
model.add(layers.Flatten())
```

```
model.add(layers.Dense(256, activation='relu'))
```

```
model.add(layers.Dropout(0.5))
```

```
model.add(layers.Dense(1, activation='sigmoid'))
```

```
base_model.trainable=False
```

```
model.compile(optimizer=tf.keras.optimizers.RMS
```

```
prop(learning_rate=2e-5)
```

```
loss='binary_crossentropy'
```

```
metrics=['accuracy'])
```

```
history = model.fit(
```

```
train_datagen
```

```
steps_per_epoch=10,
```

```
epochs=10,
```

```
validation_data=validation_generator,
```

```
validation_steps=10
```

```
)
```

```
test_loss, test_acc = model.evaluate
```

```
(validation_generator)
```

```
print('test accuracy', test_acc)
```

Output :

Predicted : Dog

import ~~not~~ numpy as np

from tensorflow import preprocessing import image

image_path = '/kaggle/input/cats-and-dogs/test/dog/americano-bull-dog-12.jpg' rf.d6fd6935cb14c99
le224x224x3

img = image.load_img(image_path, target_size=(224, 224))

img_array = image.img_to_array(img)

img_array = np.expand_dims(img_array, axis=0)

img_array / 255;

prediction = model.predict(img_array)

if prediction[0] > 0.5:

print("Predicted : Dog")

else:

print("Predicted : Cat")

- Result :

Implementation of deep learning algorithm was implemented successfully.