DS505: INTRODUCTION TO DEEP LEARNING

P04: Convolutional Neural Network (CNN)

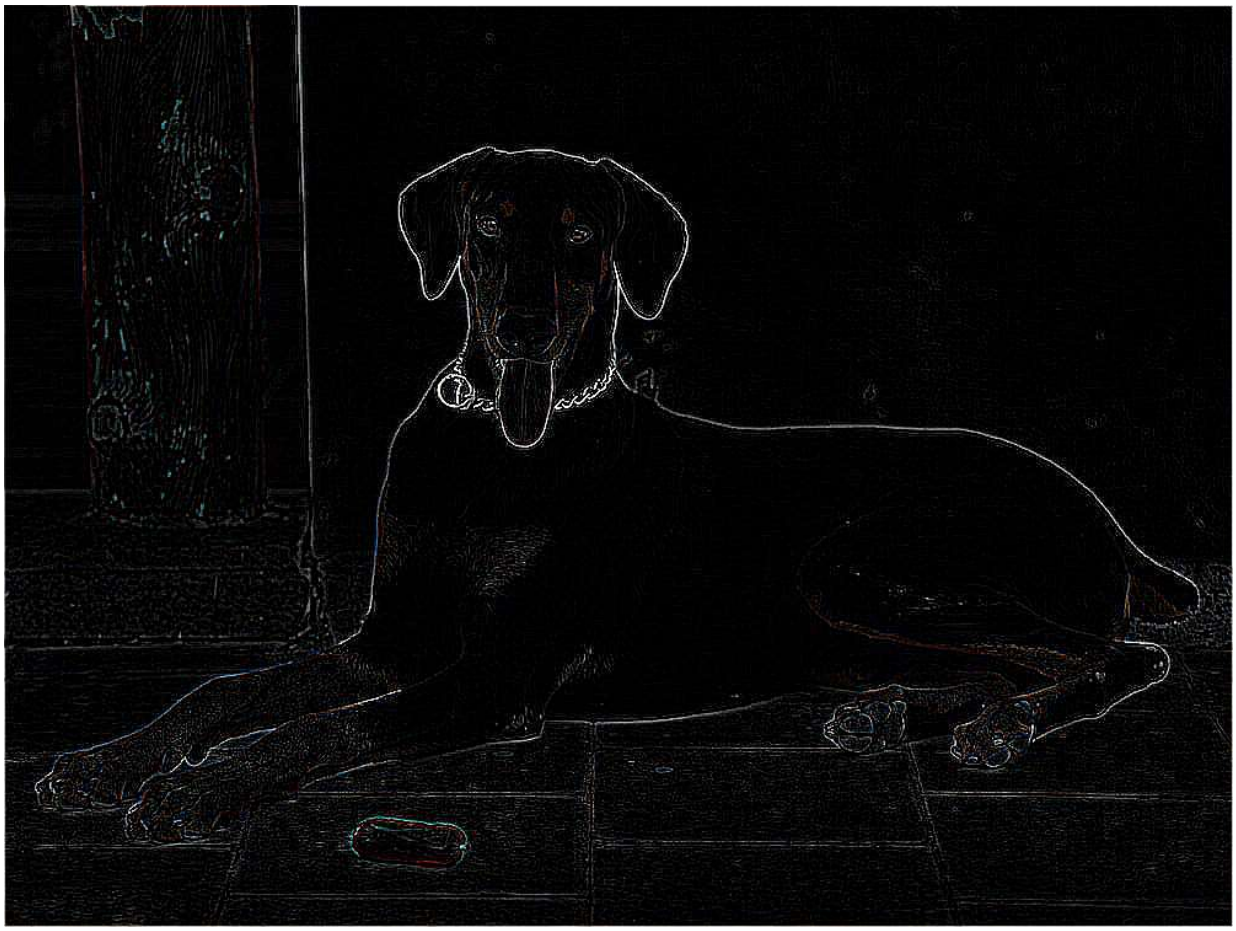
In [1]:
```
from PIL import Image, ImageFilter
```

In [3]:
```
image = Image.open('dobermann.jpg').convert('RGB')
```

In [4]:
```
display(image)
```

In [5]:
```
filtered = image.filter(ImageFilter.Kernel(size=(3,3),
                                           kernel=[-1,-1,-1,-1,8,-1,-1,-1,-1],
                                           scale=1))
```

In [6]:
```
display(filtered)
```



In [7]:
```
import tensorflow as tf
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
```

In [8]:
```
mnist = tf.keras.datasets.mnist
```

In [9]:
```
(x_train, y_train), (x_test, y_test) = mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz (https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz)
11490434/11490434 [==============================] - 4s 0us/step
```

In [10]:
```
x_train.shape, y_train.shape, x_test.shape, y_test.shape
```

Out[10]:
```
((60000, 28, 28), (60000,), (10000, 28, 28), (10000,))
```

In [11]:
```
index = 2
img = Image.fromarray(x_train[index])
print(y_train[index])
display(img)
```

```
4
```

In [12]:
```
x_train = x_train / 255.0
x_test = x_test / 255.0
```

In [13]:
```
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
```

In [14]:
```
y_train.shape
```

Out[14]:
```
(60000, 10)
```

In [15]:
```
x_train.shape
```

Out[15]:
```
(60000, 28, 28)
```

In [16]:
```
x_train = x_train.reshape(x_train.shape[0], x_train.shape[1], x_train.shape[2], 1
x_test = x_test.reshape(x_test.shape[0], x_test.shape[1], x_test.shape[2], 1)
```

In [17]:
```
x_train.shape
```

Out[17]:
```
(60000, 28, 28, 1)
```

In [18]:
```
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)))
model.add(tf.keras.layers.MaxPooling2D(pool_size=(2, 2),input_shape=(4, 4, 1)))
#model.addMaxPooling2D((pool_size=(2, 2),strides=(2, 2), padding='valid'))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(.5))
model.add(Dense(10, activation='softmax'))
```

In [19]:
```
print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 32)        320

 max_pooling2d (MaxPooling2D  (None, 13, 13, 32)       0
 )

 flatten (Flatten)           (None, 5408)              0

 dense (Dense)               (None, 128)               692352

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 10)                1290

=================================================================
Total params: 693,962
Trainable params: 693,962
Non-trainable params: 0
_________________________________________________________________
None
```

In [20]:
```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accura
model.fit(x_train, y_train, epochs=10)

model.evaluate(x_test, y_test)
```

```
Epoch 1/10
1875/1875 [==============================] - 32s 17ms/step - loss: 0.2710 - acc
uracy: 0.9179
Epoch 2/10
1875/1875 [==============================] - 32s 17ms/step - loss: 0.1081 - acc
uracy: 0.9676
Epoch 3/10
1875/1875 [==============================] - 30s 16ms/step - loss: 0.0828 - acc
uracy: 0.9754
Epoch 4/10
1875/1875 [==============================] - 28s 15ms/step - loss: 0.0644 - acc
uracy: 0.9806
Epoch 5/10
1875/1875 [==============================] - 24s 13ms/step - loss: 0.0551 - acc
uracy: 0.9830
Epoch 6/10
1875/1875 [==============================] - 27s 14ms/step - loss: 0.0461 - acc
uracy: 0.9850
Epoch 7/10
1875/1875 [==============================] - 30s 16ms/step - loss: 0.0414 - acc
uracy: 0.9866
Epoch 8/10
1875/1875 [==============================] - 31s 16ms/step - loss: 0.0364 - acc
uracy: 0.9883
Epoch 9/10
1875/1875 [==============================] - 31s 16ms/step - loss: 0.0325 - acc
uracy: 0.9898
Epoch 10/10
1875/1875 [==============================] - 30s 16ms/step - loss: 0.0304 - acc
uracy: 0.9902
313/313 [==============================] - 2s 5ms/step - loss: 0.0427 - accurac
y: 0.9878
```

Out[20]:
```
[0.042684439569711685, 0.9878000020980835]
```

In [ ]:
```
```