



*Trường Đại học Công nghệ Thông tin*  
*Khoa Khoa học máy tính*

# IMAGE CLASSIFICATION

7-Aug-20

## Phân loại hình ảnh từ vệ tinh

Giáo viên HD: ***Mai Tiến Dũng***

Sinh viên thực hiện:

***Nguyễn Duy Nhật***

***18520118***

***Đỗ Nguyễn Thuận Phong***

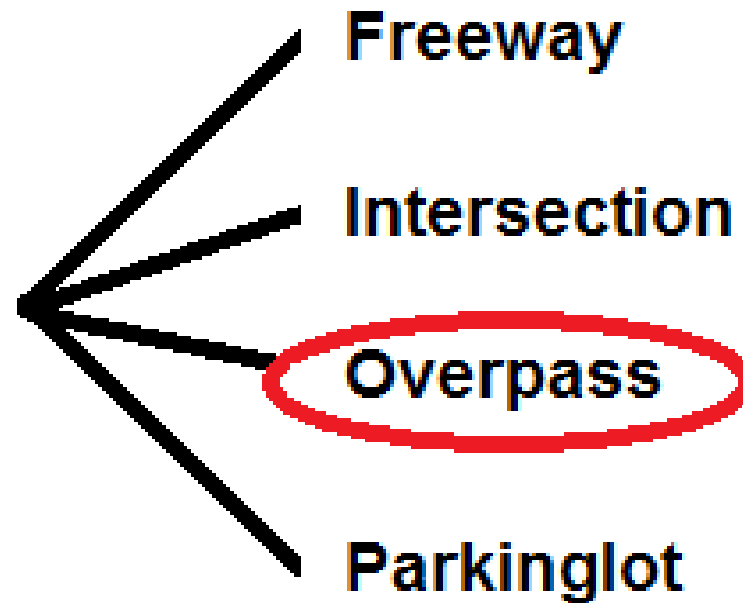
***18520126***

***Nguyễn Dương Trúc Phương***

***18520133***

# BÀI TOÁN

- Bài toán phân loại hình ảnh vệ tinh
- Input: 1 tấm hình.
- Output: Nhãn tương ứng.



# ỨNG DỤNG

- Khảo sát lập phương án quy hoạch và xây dựng giao thông đường bộ.
- Giám sát mức độ phân bố của các công trình giao thông đường bộ.
- Giúp hạn chế ùn tắc.
- Là cơ sở nền tảng để ứng dụng ảnh vệ tinh vào trong các lĩnh vực khác (như nông nghiệp, quân sự...)

# DATA

- Nguồn: <http://weegeevision.ucmerced.edu/datasets/landuse.html>



○ Freeway



○ Intersection



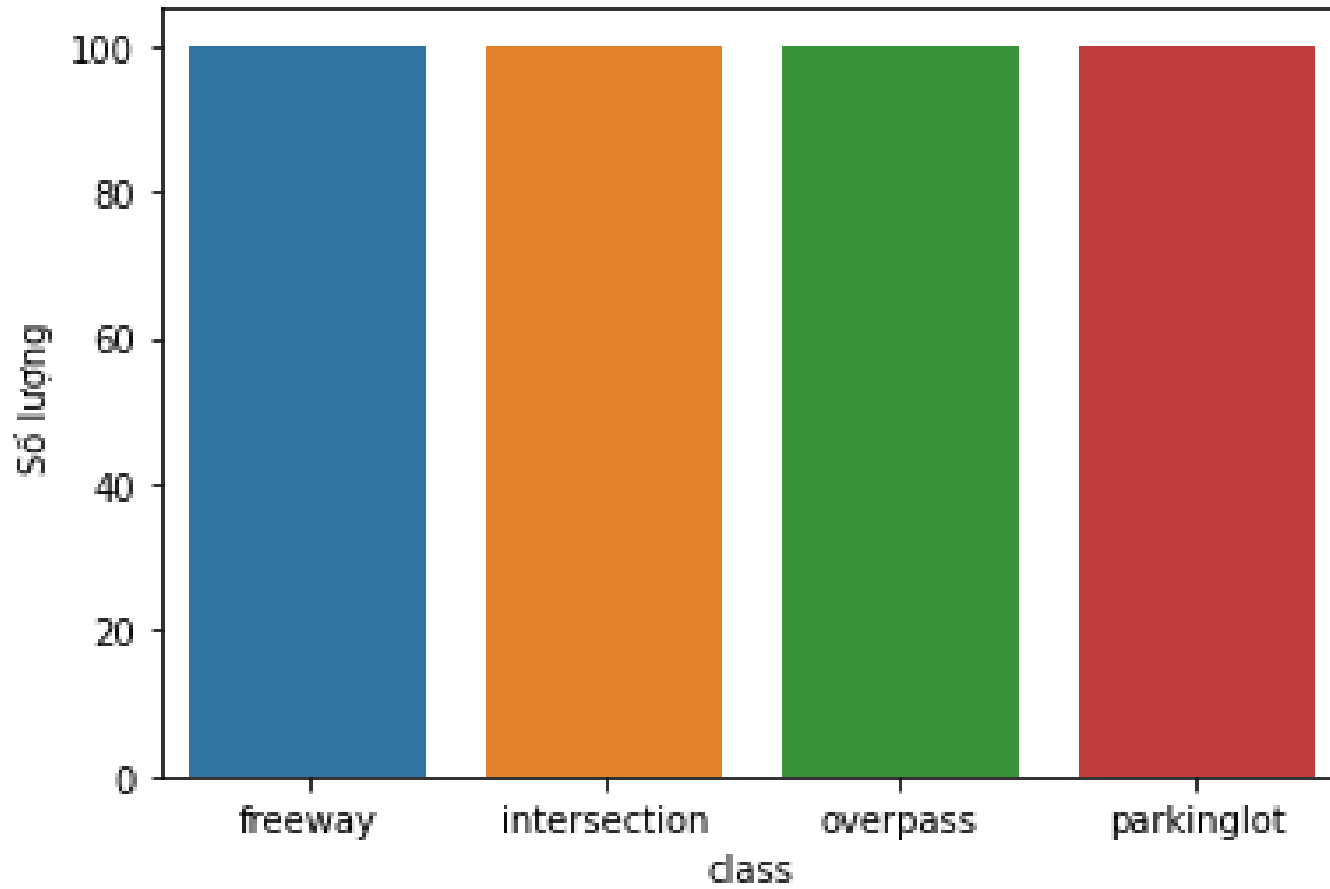
○ Overpass



○ Parkinglot

# DATA

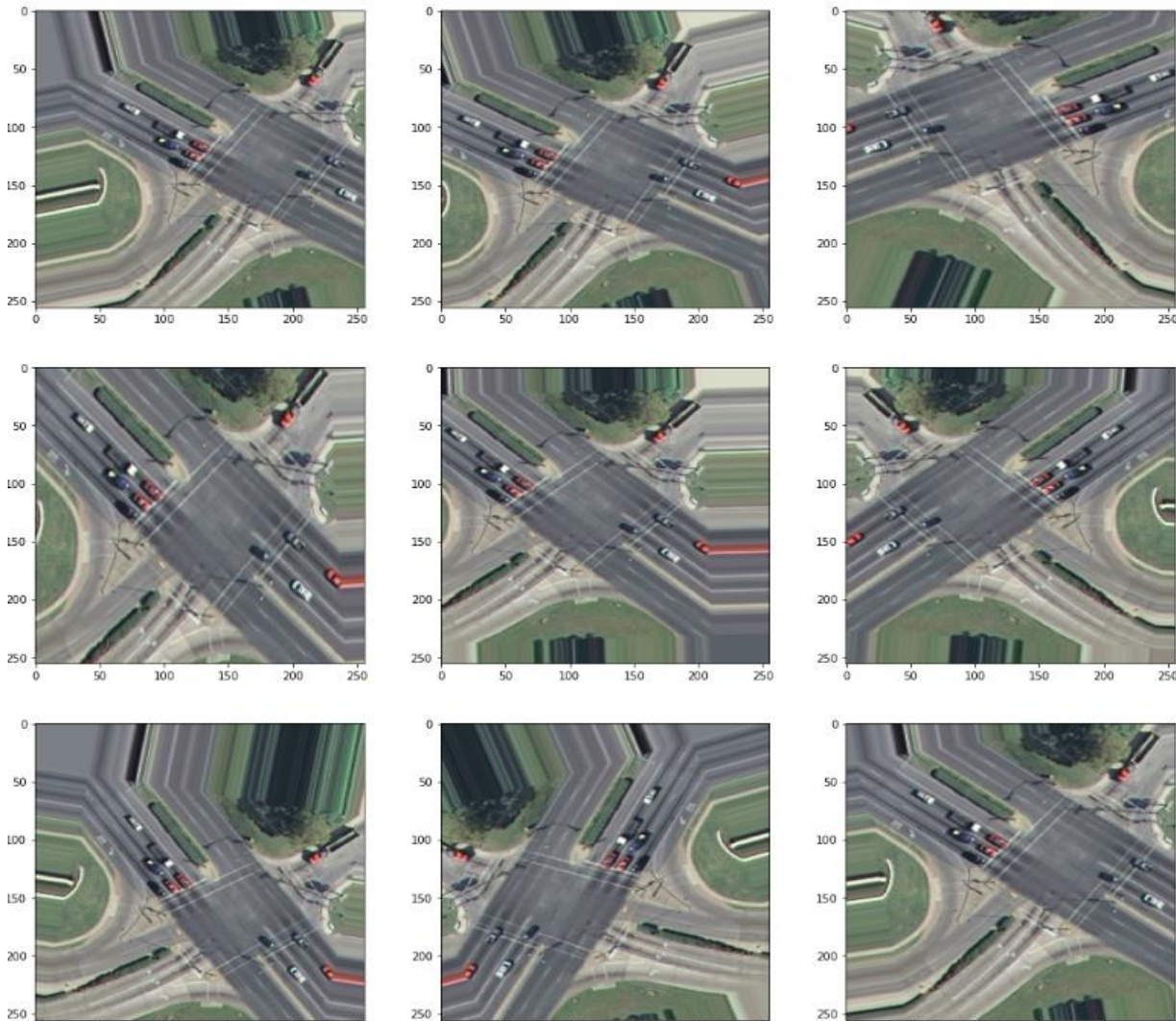
- Thống kê dữ liệu:





# DATA

- Tăng thêm data: ImageDataGenerator

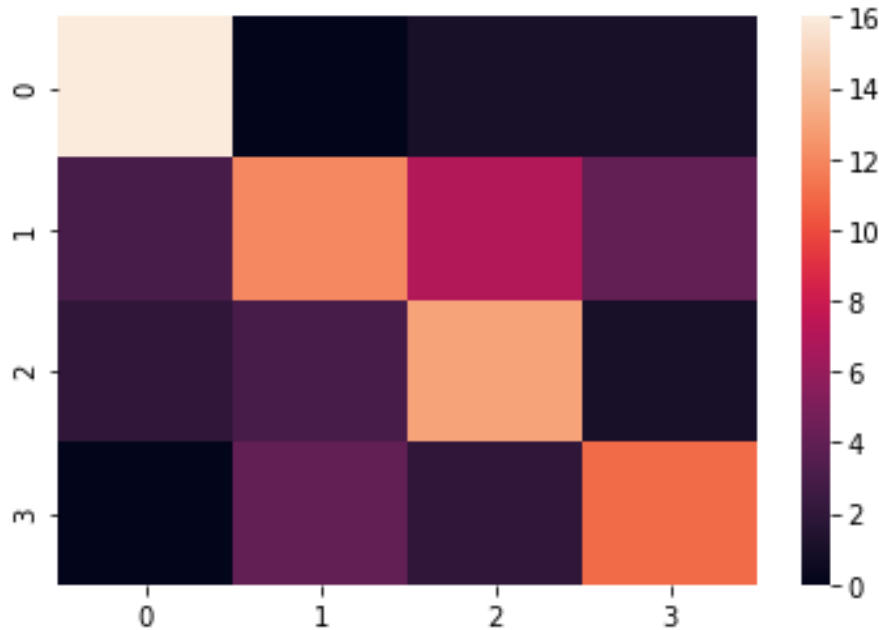


Train / Test  
80% / 20%

7-Aug-20

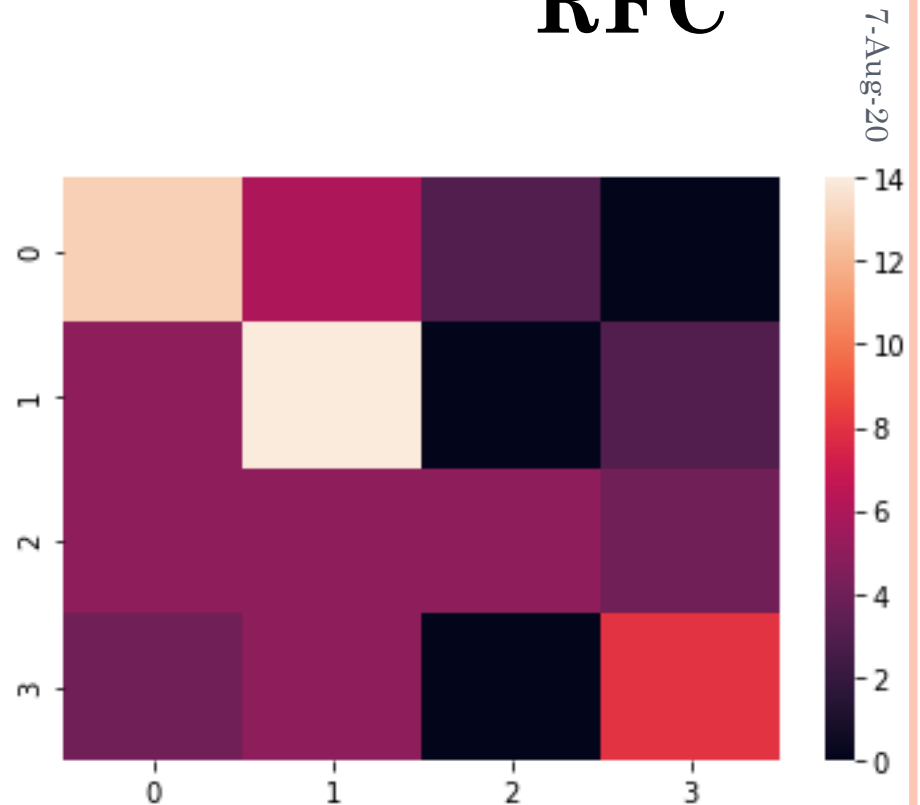
# MACHINE LEARNING

## SVM



**Accuracy: 65%**

## RFC

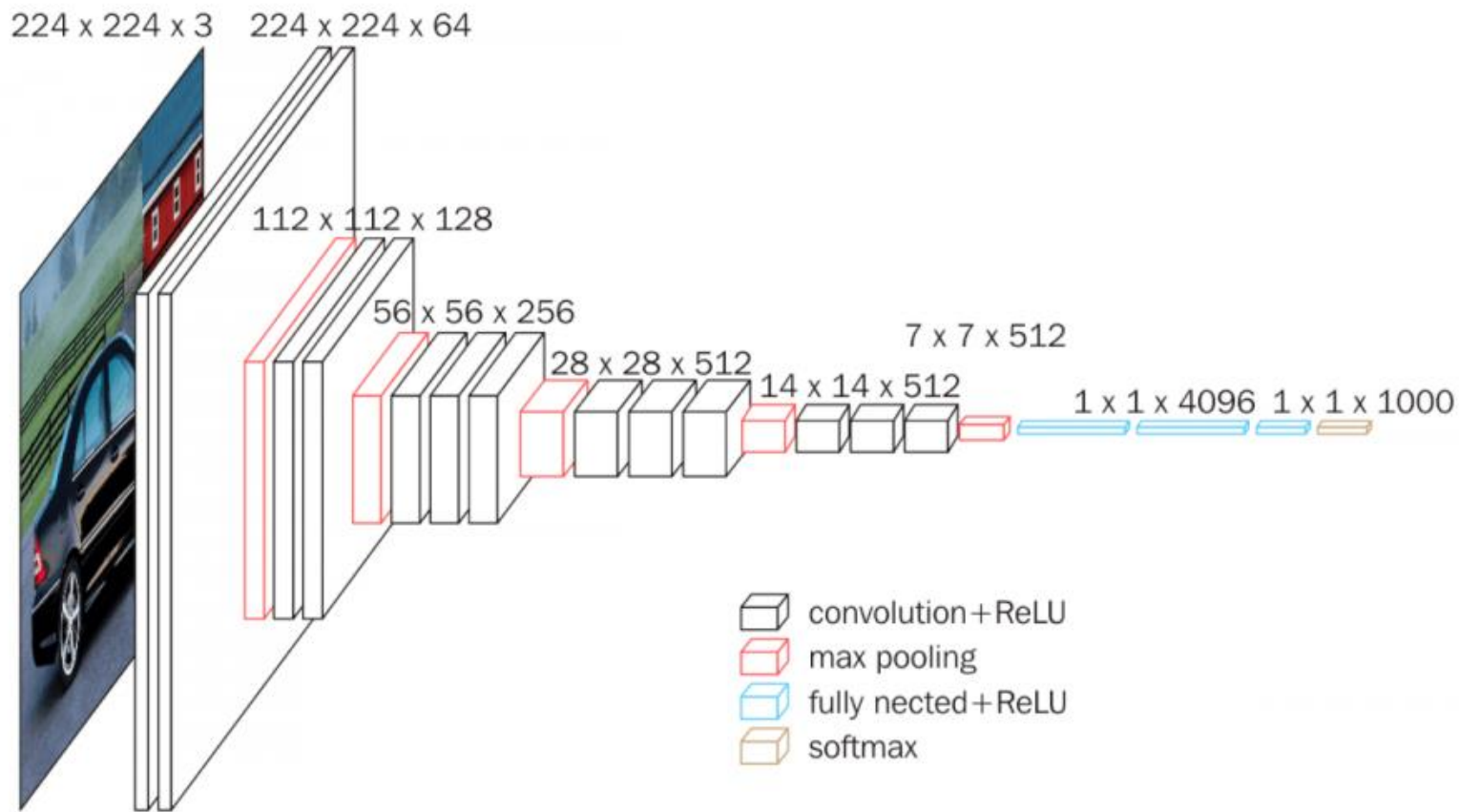


**Accuracy: 50%**

7-Aug-20

# DEEP LEARNING

- VGG16 Network: <https://arxiv.org/pdf/1409.1556.pdf>





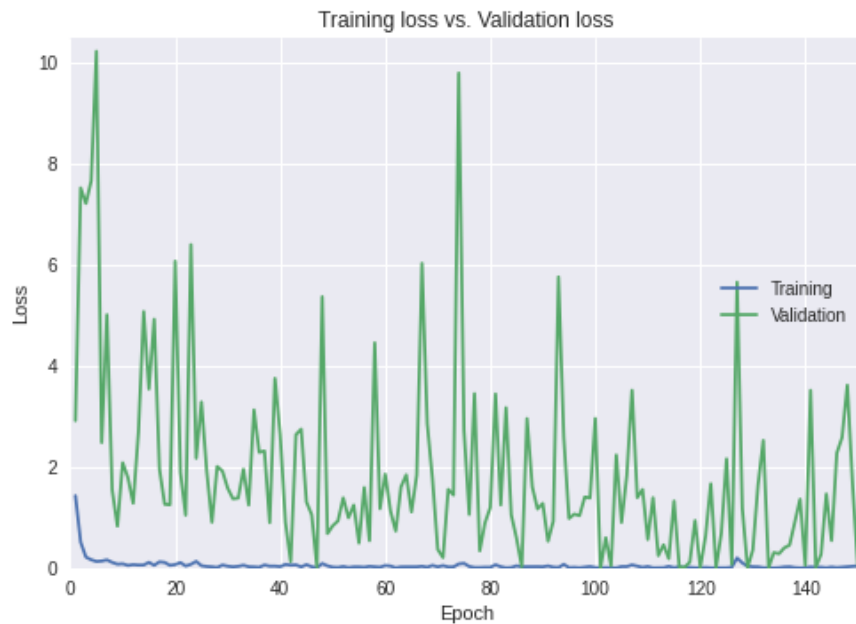
# DEEP LEARNING

## ○ Build model:

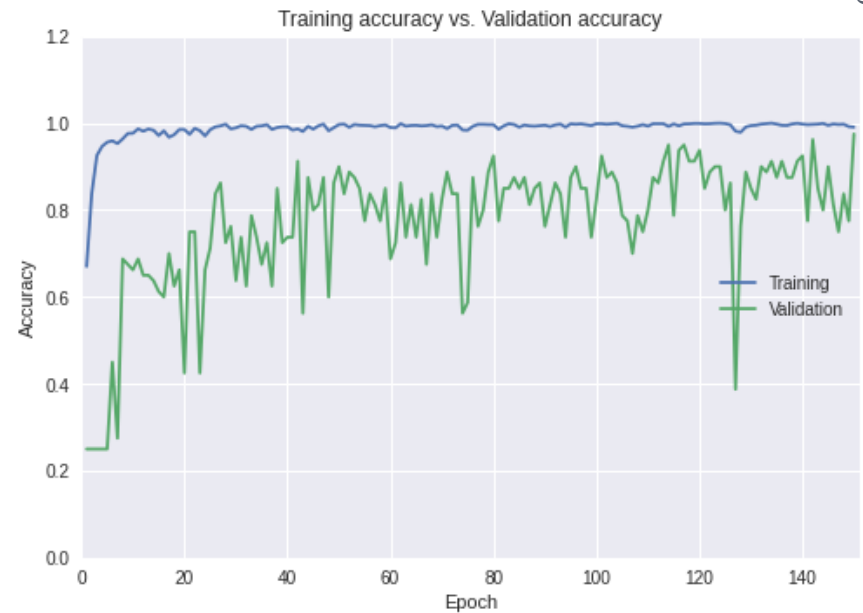
```
[ ] 1 def build_model():
2     model = Sequential()
3     model.add(Conv2D(input_shape=(224,224,3),filters=64,kernel_size=(3,3),padding="same", activation="relu"))
4     model.add(Conv2D(filters=64,kernel_size=(3,3),padding="same", activation="relu"))
5     model.add(BatchNormalization())
6     model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
7     model.add(Conv2D(filters=128,kernel_size=(3,3),padding="same", activation="relu"))
8     model.add(Conv2D(filters=128,kernel_size=(3,3),padding="same", activation="relu"))
9     model.add(BatchNormalization())
10    model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
11
12    model.add(Conv2D(filters=256,kernel_size=(3,3),padding="same", activation="relu"))
13    model.add(Conv2D(filters=256,kernel_size=(3,3),padding="same", activation="relu"))
14    model.add(Conv2D(filters=256,kernel_size=(3,3),padding="same", activation="relu"))
15    model.add(BatchNormalization())
16    model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
17
18    model.add(Conv2D(filters=512,kernel_size=(3,3),padding="same", activation="relu"))
19    model.add(Conv2D(filters=512,kernel_size=(3,3),padding="same", activation="relu"))
20    model.add(Conv2D(filters=512,kernel_size=(3,3),padding="same", activation="relu"))
21    model.add(BatchNormalization())
22    model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
23
24    model.add(Conv2D(filters=512,kernel_size=(3,3),padding="same", activation="relu"))
25    model.add(Conv2D(filters=512,kernel_size=(3,3),padding="same", activation="relu"))
26    model.add(Conv2D(filters=512,kernel_size=(3,3),padding="same", activation="relu"))
27    model.add(BatchNormalization())
28    model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
29
30    model.add(Flatten())
31    model.add(Dense(units=4096,activation="relu"))
32    model.add(Dense(units=4096,activation="relu"))
33    model.add(Dropout(0.5))
34    model.add(Dense(units=4,activation="softmax"))
35    return model
```

# DEEP LEARNING

## Loss



## Accuracy



7-Aug-20

# DEMO

```
[ ] 1 img=cv2.imread('/content/drive/My Drive/Colab Notebooks/ML/ImageClassification/UCM
    2 cv2_imshow(img)
    3 img=cv2.resize(img,(224,224))
    4 img=img/255.0
    5 img = np.expand_dims(img, axis=0)
    6 result=saved_model.predict(img)
    7 print(select_label(result))
```



Freeway

**Freeway**

# DEMO

```
[ ] 1 img=cv2.imread('/content/drive/My Drive/Colab Notebooks/ML/ImageClassification/UCMerced_LandUse/Images/d
2 cv2.imshow(img)
3 img=cv2.resize(img,(224,224))
4 img=img/255.0
5 img = np.expand_dims(img, axis=0)
6 result=saved_model.predict(img)
7 print(select_label(result))
```



Intersection

Intersection

7-Aug-20

12

# DEMO

```
[ ] 1 img=cv2.imread('/content/drive/My Drive/Colab Notebooks/ML/ImageClassification/UCMerced_LandUse/  
2 cv2_imshow(img)  
3 img=cv2.resize(img,(224,224))  
4 img=img/255.0  
5 img = np.expand_dims(img, axis=0)  
6 result=saved_model.predict(img)  
7 print(select_label(result))
```

7-Aug-20



Overpass

**Overpass**



# DEMO

```
[ ] 1 img=cv2.imread('/content/drive/My Drive/Colab Notebooks/ML/ImageClas
2 cv2_imshow(img)
3 img=cv2.resize(img,(224,224))
4 img=img/255.0
5 img = np.expand_dims(img, axis=0)
6 result=saved_model.predict(img)
7 print(select_label(result))
```



Parkinglot

**Parkinglot**



