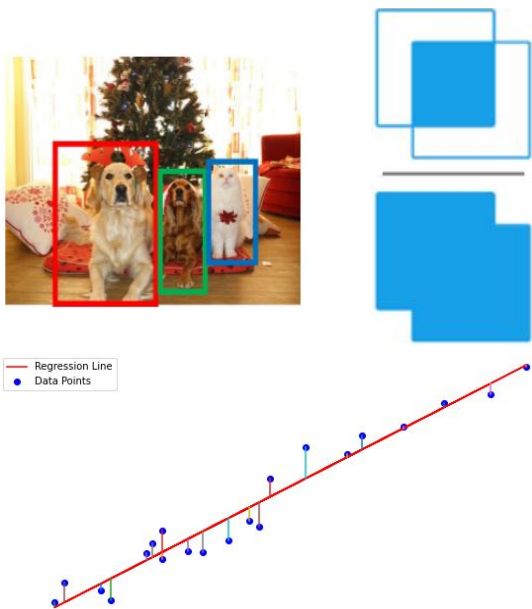


Competition tasks and metrics



Google
colab

kaggle

1. Working and Learning with Colab
2. Organizing Data
3. Evaluation metric
4. Metrics for regression
5. Metrics for classification
6. Metrics for Object Detection

1 - Working and Learning with Colab

Kaggle & Kaggle Notebooks

- 12h sử dụng CPU/GPU
- 9h sử dụng TPU
- 20 gb dung lượng
- Lưu tự động vào workspace
- Không gian lưu trữ bên ngoài (Không lưu tự động vào workspace)

Thông Số CPU:

- 4 CPU cores
- 16 gb RAM

Thông số GPU

- 2 CPU cores
- 13 gb RAM

Thông số TP

- 4 CPU cores
- 16 gb RAM

The screenshot displays the Kaggle homepage. At the top, there is a search bar labeled 'Search competitions' and a 'Filters' button. Below this, a row of category buttons includes 'All Competitions' (Everything, past & present), 'Featured' (Premier challenges with prizes), 'Getting Started' (Approachable ML fundamentals), 'Research' (Scientific and scholarly challenges), 'Community' (Created by fellow Kagglers), and 'Playground' (Fun practice problems). The main section is titled 'Active Competitions' with a 'Hotness' dropdown menu. It features a grid of competition cards. The first card is 'Vesuvius Challenge - Ink Detection' with a \$1,000,000 prize and a deadline of 'a month to go'. The second is 'Google - American Sign Language Fingerspelling...' with a \$200,000 prize and '3 months to go'. The third is 'Parkinson's Freezing of Gait Prediction' with a \$100,000 prize and 'a month to go'. The fourth is '2023 Kaggle AI Report' with a \$70,000 prize and '2 months to go'. Each card includes a thumbnail image, the competition title, a brief description, the prize amount, and the time remaining.

2 - Organizing Data

1- Setting up a dataset

Menu_Dataset-1 (Public)
ig/Menu_Dataset

Issues Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

This branch is up to date with dotrannhattuong/Menu_Dataset:main. Contribute Sync fork

dotrannhattuong Update label menu sentences 846e156 on Dec 26, 2022 5 commits

config	Upload Data	8 months ago
dataset_det	Upload Data	8 months ago
dataset_rec	Upload Total VinAI Dataset	5 months ago
dictionary	Upload Data	8 months ago
label_sentences	Update label menu sentences	5 months ago
notebook	Upload Data	8 months ago
README.md	Upload Total VinAI Dataset	5 months ago

datasetOCR (Public)
y/datasetOCR

Issues Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

This branch is 1 commit behind thangquocdang:main. Contribute Sync fork

thangquocdang add meta 64af384 on Aug 8, 2022 7 commits

InkData_line_processed	add InkData_line_processed	9 months ago
en_00	add en_00	9 months ago
en_01	add en_01	9 months ago
meta	add meta	9 months ago
random	add random	9 months ago
vi_00	add vi_00	9 months ago
vi_01	add vi_01	9 months ago

small_dog_cat_dataset (Public)

Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

Go to file Add file Code

anminhhung for fun 8477eb1 on Nov 15, 2022 1 commit

test	for fun	6 months ago
train	for fun	6 months ago

vehicle_data (Public)

Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

Go to file Add file Code

anminhhung first commit b66f524 on Jul 1, 2022 1 commit

train	first commit	last year
val	first commit	last year

2 - Organizing Data

2- Working with Data

```
import matplotlib.pyplot as plt
import cv2 # thư viện xử lý ảnh

# đọc ảnh từ đường dẫn
image = cv2.imread("small_dog_cat_dataset/test/cats/cat.10028.jpg")

plt.figure(figsize=(6, 6))
plt.imshow(image)
plt.show()
```



```
import matplotlib.pyplot as plt
import cv2 # thư viện xử lý ảnh

# đọc ảnh từ đường dẫn
image = cv2.imread("small_dog_cat_dataset/test/cats/cat.10028.jpg")
rgb_image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

plt.figure(figsize=(6, 6))
plt.imshow(rgb_image)
plt.show()
```



2 - Organizing Data

2- Working with Data

```
import matplotlib.pyplot as plt
import cv2
import glob

list_image_path = glob.glob("small_dog_cat_dataset/test/cats/*.jpg")
list_image = []

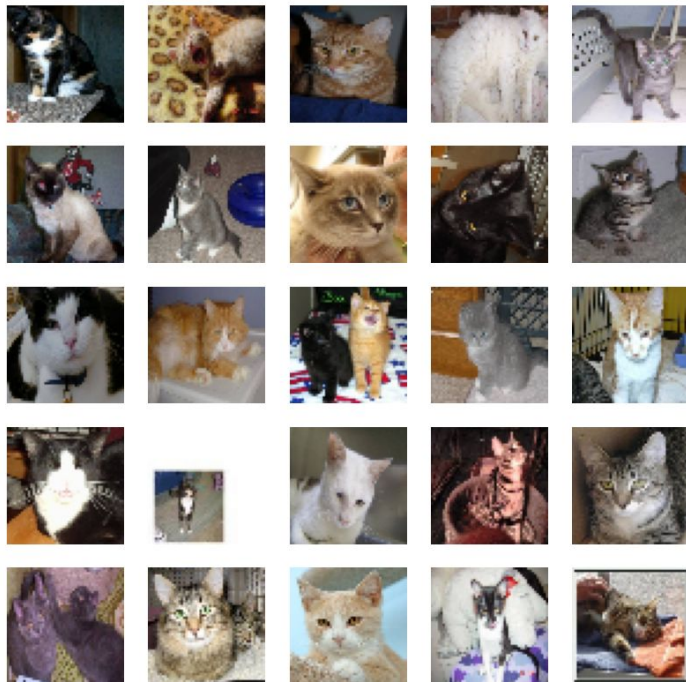
for i in range(25):
    image_path = list_image_path[i]
    image = cv2.imread(image_path)
    rgb_image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    # resize image
    rgb_image = cv2.resize(rgb_image, (50, 50))

    list_image.append(rgb_image)

fig = plt.figure(figsize=(10, 10))
# define width and height
columns = 5
rows = 5

for i in range(1, columns*rows + 1):
    fig.add_subplot(rows, columns, i)
    plt.axis('off') # tắt hiển thị 2 trục
    plt.imshow(list_image[i-1])
```



2 - Organizing Data

2- Working with Data

```
import glob

list_cats_test = glob.glob("small_dog_cat_dataset/test/cats/*.jpg")
list_dogs_test = glob.glob("small_dog_cat_dataset/test/dogs/*.jpg")

print(len(list_cats_test), len(list_dogs_test))

list_cats_train = glob.glob("small_dog_cat_dataset/train/cats/*.jpg")
list_dogs_train = glob.glob("small_dog_cat_dataset/train/dogs/*.jpg")

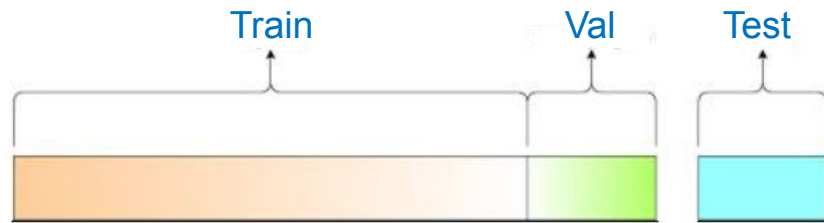
print(len(list_cats_train), len(list_dogs_train))

new_list_cats_train = list_cats_train[:800]
new_list_dogs_train = list_dogs_train[:800]

list_cats_val = list_cats_train[800:]
list_dogs_val = list_dogs_train[800:]

print("train: ", len(new_list_cats_train), len(new_list_dogs_train))
print("val: ", len(list_cats_val), len(list_dogs_val))
print("test: ", len(list_cats_test), len(list_dogs_test))
```

300 300
1000 1000
train: 800 800
val: 200 200
test: 300 300



3 - Evaluation metric



Submission file



Leaderboard [Raw Data](#) [Refresh](#)

Search leaderboard

Public Private

This leaderboard is calculated with approximately 44% of the test data. The final results will be based on the other 56%, so the final standings may be different.

■ Prize Contenders

#	Team	Members	Score	Entries	Last	Join
1	So Tricky			0.441	92	15h
2	whatever			0.440	145	10h
3	barf			0.425	29	16h
4	I & I & R			0.415	155	7h
5	chenchen			0.408	36	6h
6	Oh no			0.404	90	2h
7	OMG			0.395	78	14h
8	loftglue			0.385	7	1d
9	RMD-3DV			0.363	23	2d

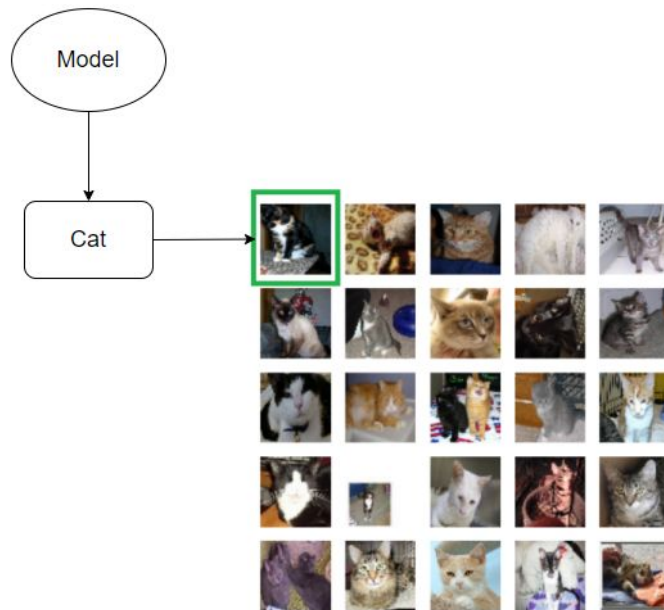
Leaderboard thể hiện kết quả của evaluation metric

Sự liên kết giữa submission file và evaluation metric rất quan trọng, bởi vì chúng ta cần xem xét sự hiệu quả của model sau khi training thông qua 2 thông tin này

3 - Evaluation metric

1- Loss Function

```
"""
y_pred là kết quả dự đoán của model
y_hat là ground truth (nhãn) của điểm dữ liệu
"""
def base_loss_function(y_pred, y_hat):
    if y_pred == y_hat:
        return 0
    else:
        return 1
```



Loss Function được định nghĩa trên 1 điểm dữ liệu bằng việc xét kết quả dự đoán của model và ground truth của điểm dữ liệu đó để tính độ lỗi.

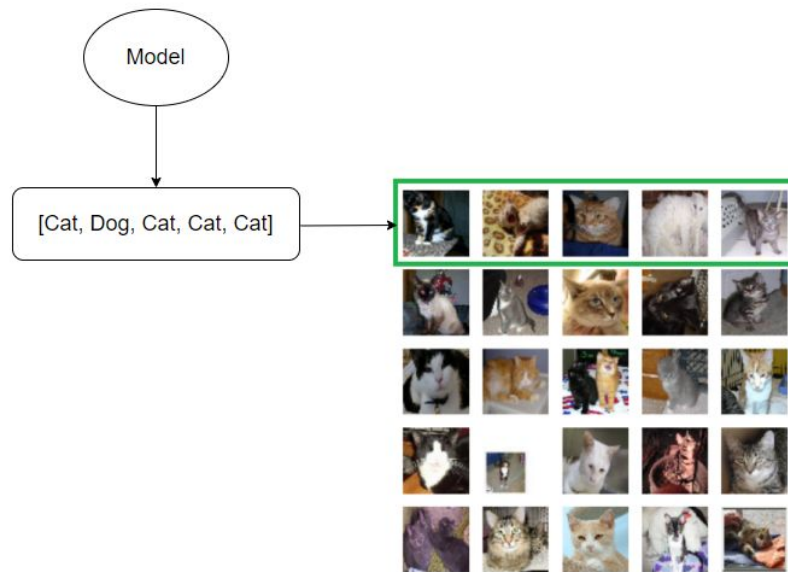
3 - Evaluation metrics

2- Cost Function

```
"""
y_pred là kết quả dự đoán của model
y_hat là ground truth (nhãn) của điểm dữ liệu
"""
def base_loss_function(y_pred, y_hat):
    if y_pred == y_hat:
        return 0
    else:
        return 1

"""
list_y_pred là danh sách kết quả dự đoán của model
list_y_hat là danh sách ground truth (nhãn) của điểm dữ liệu
"""
def base_cost_function(list_y_pred, list_y_hat):
    loss = 0
    for y_pred, y_hat in zip(list_y_pred, list_y_hat):
        loss += base_loss_function(y_pred, y_hat)

    return loss
```



Tính toán trên toàn bộ dữ liệu (hoặc một phần từ tập dữ liệu) tính toán tổng hoặc trung bình độ lỗi của các điểm dữ liệu đang xét

3 - Evaluation metric

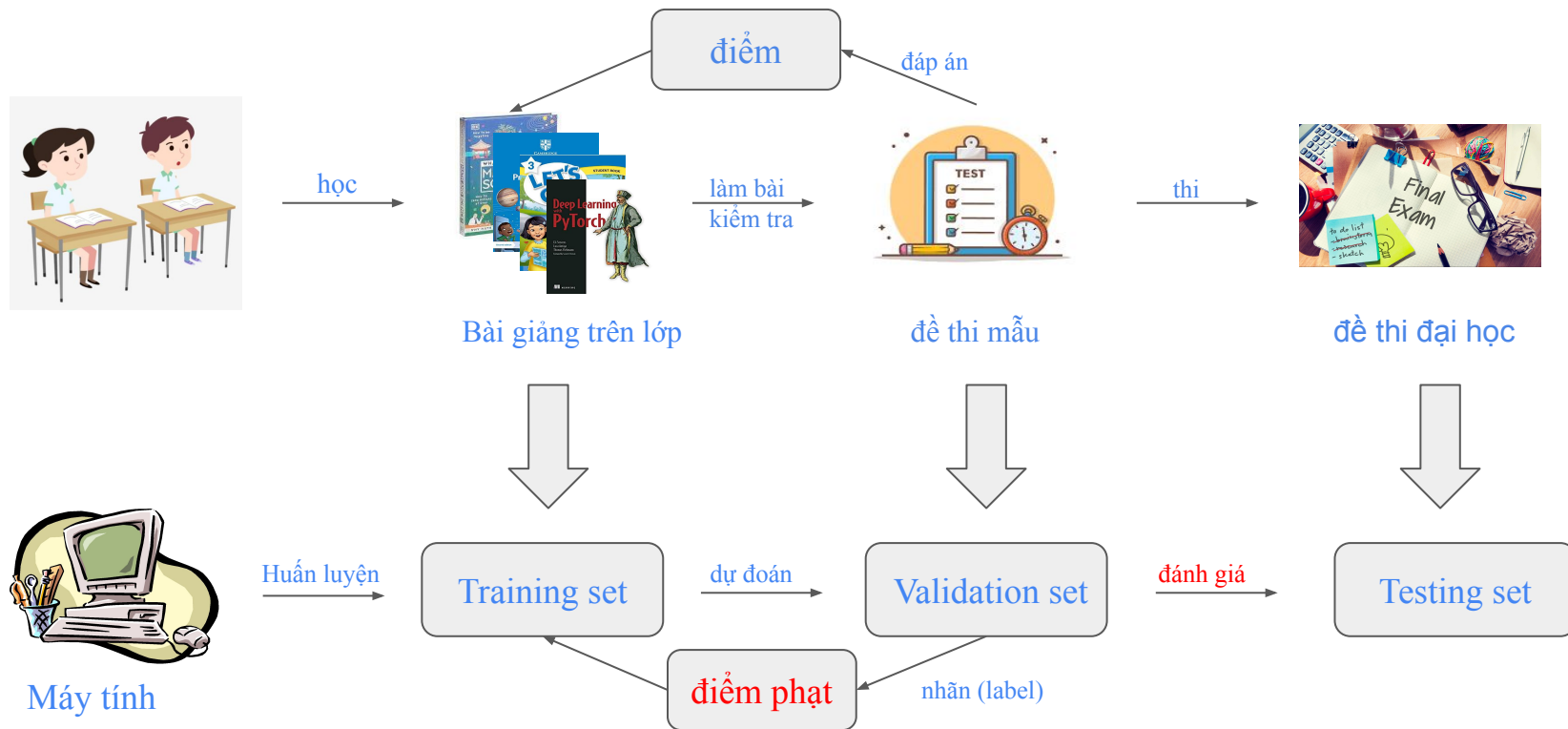
3- Objective Function & Evaluation metric

Objective Function: “is the **most general** (and safe-to-use) term **related** to the **scope** of **optimization during machine learning training**: it **comprises cost functions**, but it is **not limited** to them” - Kaggle book

Objective Function	Evaluation metric
- Tham gia vào quá trình training (error minimization / score maximization)	- Không tham gia vào quá trình training (cung cấp score sau quá trình training)
- Ảnh hưởng đến quá trình học của model	- Không ảnh hưởng đến quá trình học của model

3 - Evaluation metric

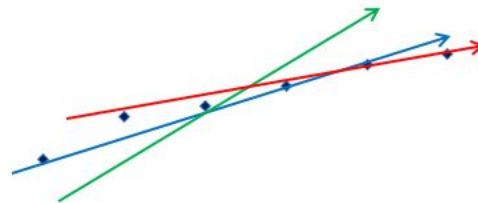
3- Objective Function & Evaluation metric



3 - Evaluation metrics

4- Basic type of tasks - Regression

# price	# area	# bedrooms	# bathrooms	# stories
Price of the Houses	Area of a House	Number of House Bedrooms	Number of House Bathrooms	Number of House Stories
				
1.75m 13.3m	1650 16.2k	1 6	1 4	1 4
13300000	7420	4	2	3
12250000	8960	4	4	4
12250000	9960	3	2	2
12215000	7500	4	2	2
11410000	7420	4	1	2
10850000	7500	3	3	1
10150000	8500	4	3	4
10150000	16200	5	3	2
9870000	8100	4	1	2
9800000	5750	3	2	4
9800000	13200	3	1	2



Housing Prices Dataset

Regression yêu cầu xây dựng một model có thể dự đoán một real-number (thông thường là một số dương)

3 - Evaluation metrics

4- Basic type of tasks - Image Classification



Dog: 0.91
Cat: 0.18

Image Classification yêu cầu xây dựng một model có thể dự đoán **nhãn** của một ảnh

3 - Evaluation metrics

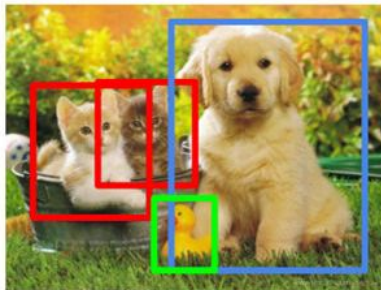
4- Basic type of tasks - Object Detection

Classification



CAT

Object Detection



CAT, DOG, DUCK

Object Detection yêu cầu xây dựng một model có thể dự đoán **vị trí** của đối tượng trong ảnh và tại vị trí đó xác định đối tượng đó là gì.

4 - Metrics for regression

1- Mean Square Error (MSE)

```
import numpy as np

def my_mean_squared_error(act, pred):
    differences_squared = (pred - act) ** 2
    mean_diff = differences_squared.mean()

    return mean_diff

act = np.array([1, 2, 3])
pred = np.array([1.1, 2.3, 3.7])

my_mean_squared_error(act, pred)
```

$$MSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}$$

Mean Square Error được tính bằng trung bình cộng của bình phương chênh lệch giữa kết quả dự đoán và kết quả thực tế.

4 - Metrics for regression

2- Root Mean Square Error (RMSE)

```
import numpy as np

def root_mean_squared_error(act, pred):
    differences_squared = (pred - act) ** 2
    mean_diff = differences_squared.mean()
    rmse_val = np.sqrt(mean_diff)

    return rmse_val

act = np.array([1, 2, 3])
pred = np.array([1.1, 2.3, 3.7])

root_mean_squared_error(act,pred)
```

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(y_i - \hat{y}_i)^2}{n}}$$

Root Mean Square Error được tính là căn bậc hai của MAE.

4 - Metrics for regression

3- Mean Absolute Error (MAE)

```
import numpy as np

def my_mean_absolute_error(act, pred):
    abs_diff = np.absolute(pred - act)
    mean_diff = abs_diff.mean()

    return mean_diff

act = np.array([1, 2, 3])
pred = np.array([1.1, 2.3, 3.7])

my_mean_absolute_error(act,pred)
```

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Mean Absolute Error được tính bằng giá trị trung bình của chênh lệch tuyệt đối giữa kết quả dự đoán và kết quả thực tế.

5 - Metrics for classification

1- Accuracy

```
import numpy as np

def acc(y_true, y_pred):
    correct = np.sum(y_true == y_pred)
    return float(correct) / y_true.shape[0]
```

$$acc = \frac{CorrectAnswers}{TotalAnswers}$$

Là tỉ lệ giữa số điểm dữ liệu được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm thử.

5 - Metrics for classification

2- Confusion matrix

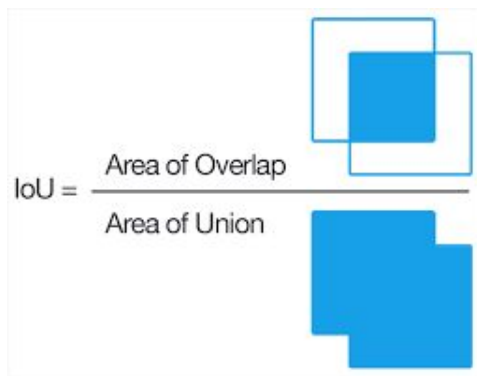
```
def my_confusion_matrix(y_true, y_pred):  
    N = np.unique(y_true).shape[0] # number of classes  
    cm = np.zeros((N, N))  
  
    for n in range(y_true.shape[0]):  
        cm[y_true[n], y_pred[n]] += 1  
  
    return cm  
  
my_confusion_matrix(y_true, y_pred)
```

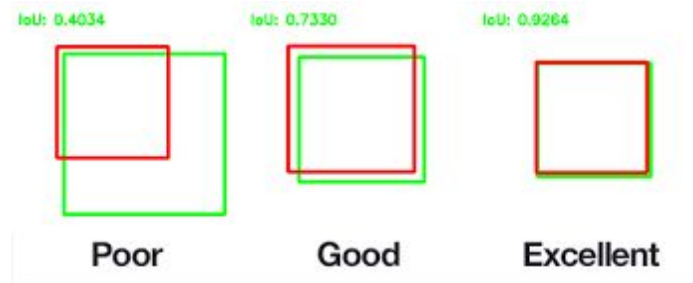
Actual Label

	A	B	C
A	856 28.98%	58 1.96%	130 4.4%
B	0	765 25.90%	136 4.6%
C	69 2.34%	33 1.12%	907 30.7%

6 - Metrics for object detection

Intersection Over Union (IOU) là chỉ số đánh giá độ chính xác của model Object Detection trên một bộ dữ liệu cụ thể. Được xác định bởi tỷ lệ giữa vùng “overlap” và vùng “combine” giữa kết quả dự đoán của model và Ground Truth


$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



IOU là một metric phù hợp để đánh giá model khi:

- Dự đoán là sai khi model không dự đoán được đối tượng bên trong vùng Ground Truth
 - Kết quả dự đoán tràn (overflow) khỏi vùng Ground Truth