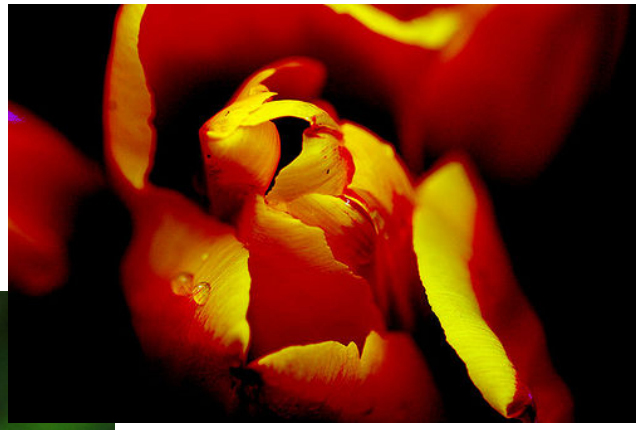
CSCI218: Foundations of Artificial Intelligence

UNIVERSITY
OF WOLLONGONG
AUSTRALIA

Flowers Recognition (KNN, MLP and CNN)

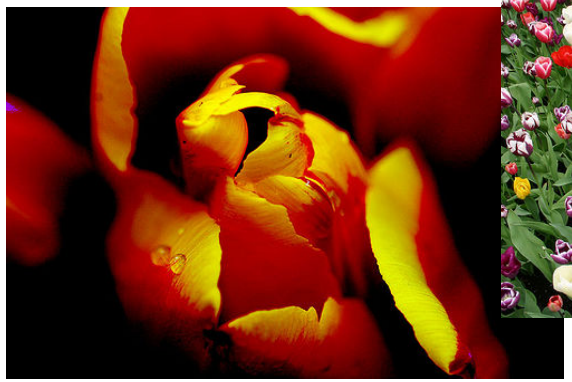


Outline

- Background
- Dataset
- Task
- Approach

Background

- Several available apps which can identify flowers and plants, such as Microsoft's Bing and Google plant identifier, Google lens.
- Automation of flower classification can boost flower searching for public and floriculture, especially in special scenario, like digital library.
- Large intra-class variation and small inter-class variation among different classes of flower makes this a challenging task.





Dataset – flower images

- This dataset contains 4317 images of flowers, based on data flicr, google images, Yandex images.
- Here are 5 classes: daisy, tulip, rose, sunflower, dandelion
- About 800~1000+ photos for each class with various pixels resolution and proportions.

Machine Learning -- KNN

- Import modules
- Extract Image features -- Color Histogram approach
- Divide dataset into training set, validation set, and test set
- Building a classifier with KNN algorithm
- Evaluation on test set by performance metrics

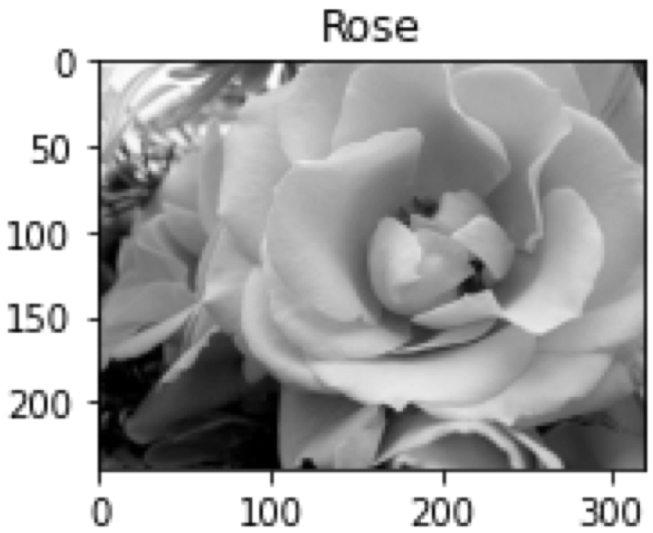
Modules

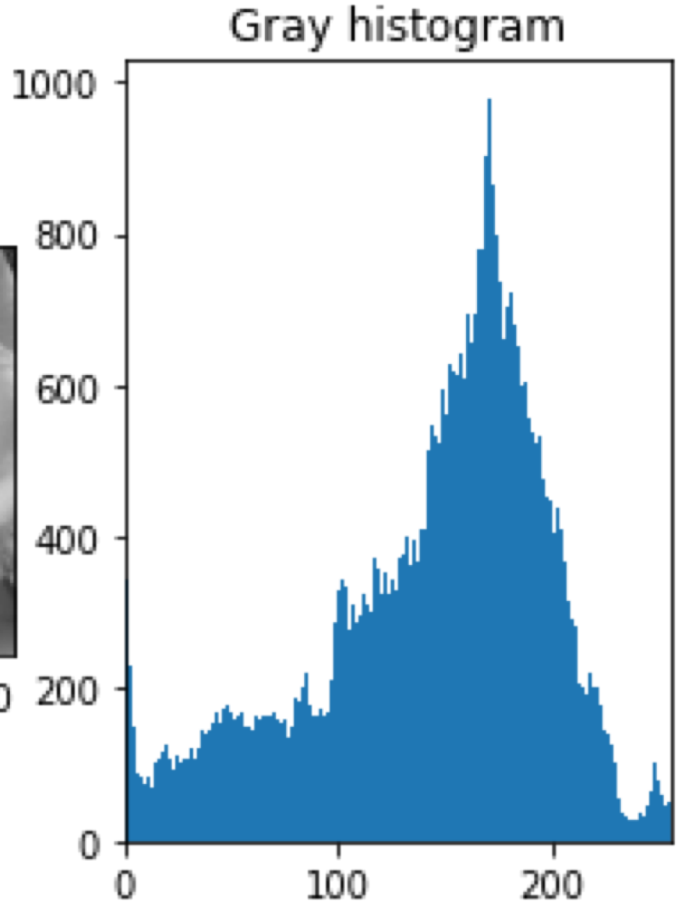
- Data visualization
  - Matplotlib
  - Seaborn
- Data processing
  - CV2 (opencv-python)
  - Numpy (Python library for working with arrays)
- Machine Learning libraries
  - Scikit-Learn
- Model performance metrics
  - Scikit-Learn

Image feature

- Color Histogram

  A plot with pixel values (ranging from 0 to 255, not always) in X-axis and corresponding number of pixels in the image on Y-axis.



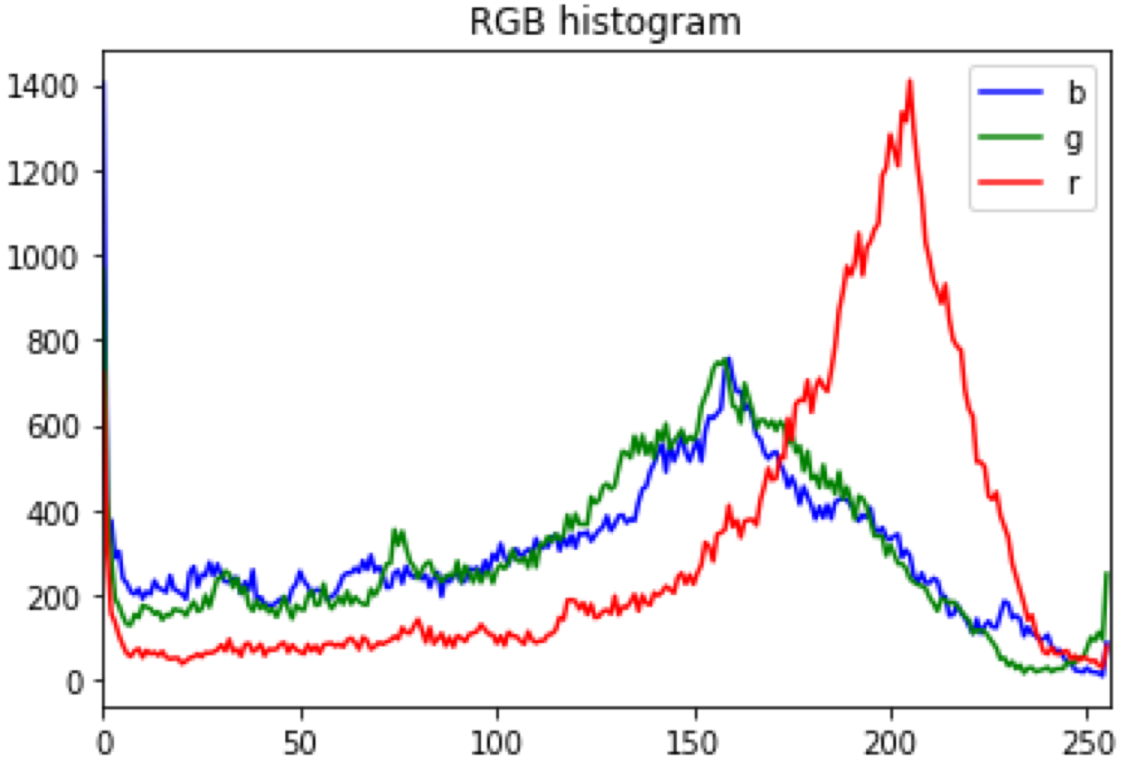


Left region of histogram shows the number of darker pixels in image and right region shows the number of brighter pixels.

This histogram is drawn for grayscale image.

Image feature

- RGB Histogram
  - **an acronym for** "Red Green Blue”
    As a colour model representing color.
- A digital image has a red, green and blue channel.
- A grayscale image has just one channel.



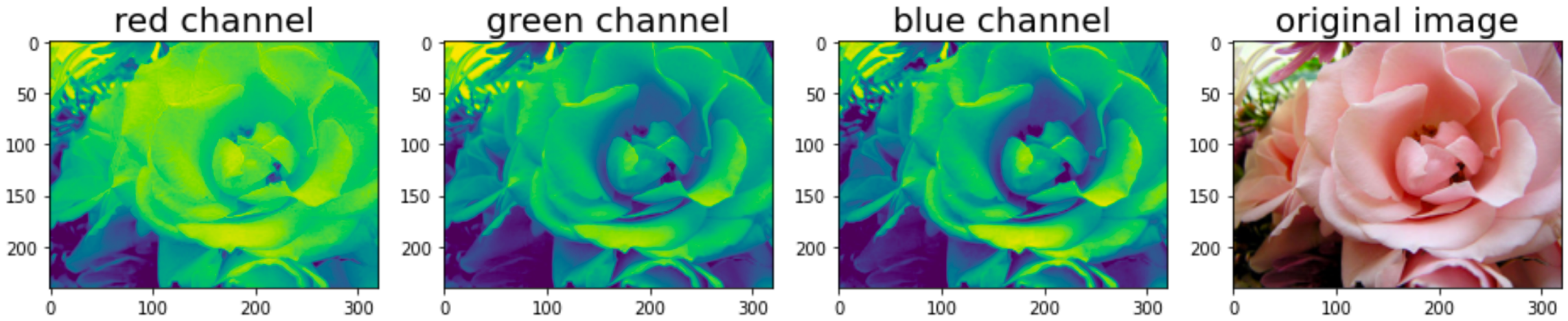


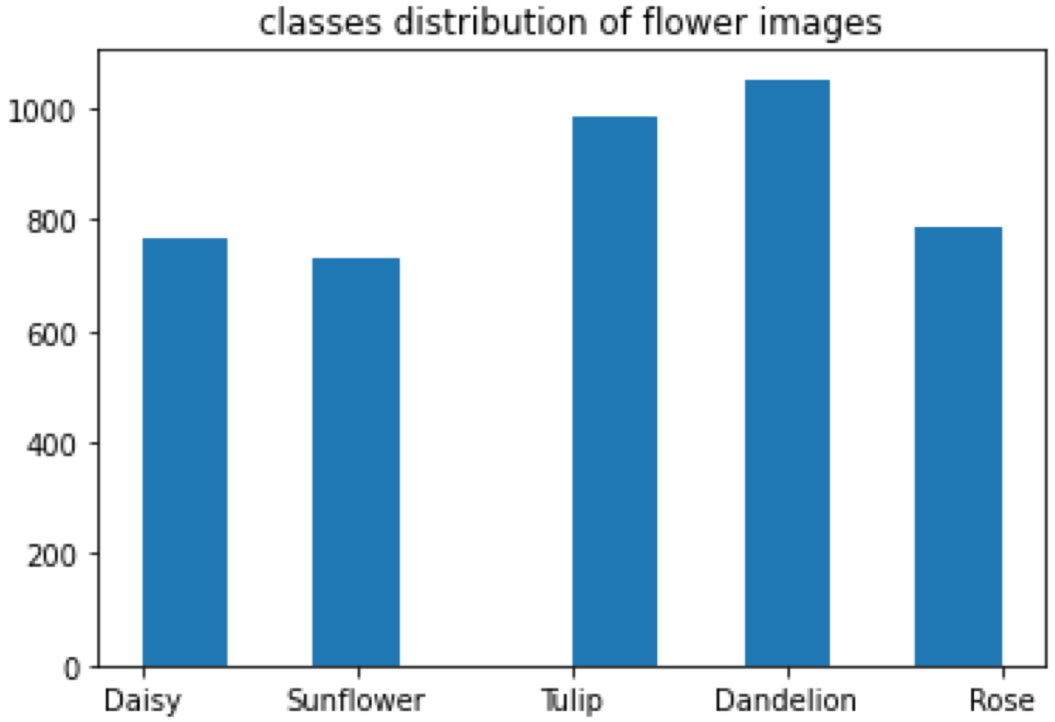
Image feature

- Function – extract_color_historgram()
  - Loading image
  - Resize image as unified size 150*150 with three channels
  - Extract from an image (size of [150, 150, 3]) to obtain a color histogram (size of [6, 6, 6])
  - Normalize the elements in histogram (scaling between 0 and 1)

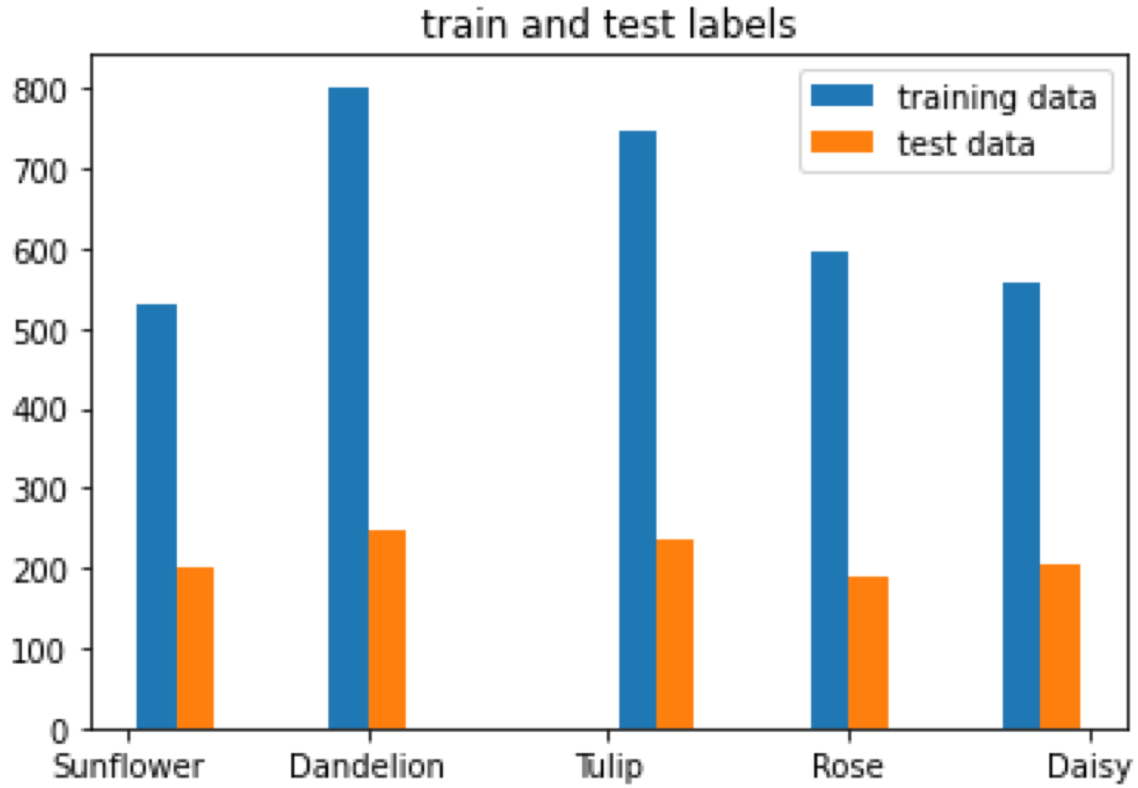
- Loading data – make_train_data

Split the data into training, validation and test sets

- Before splitting data



- After data split

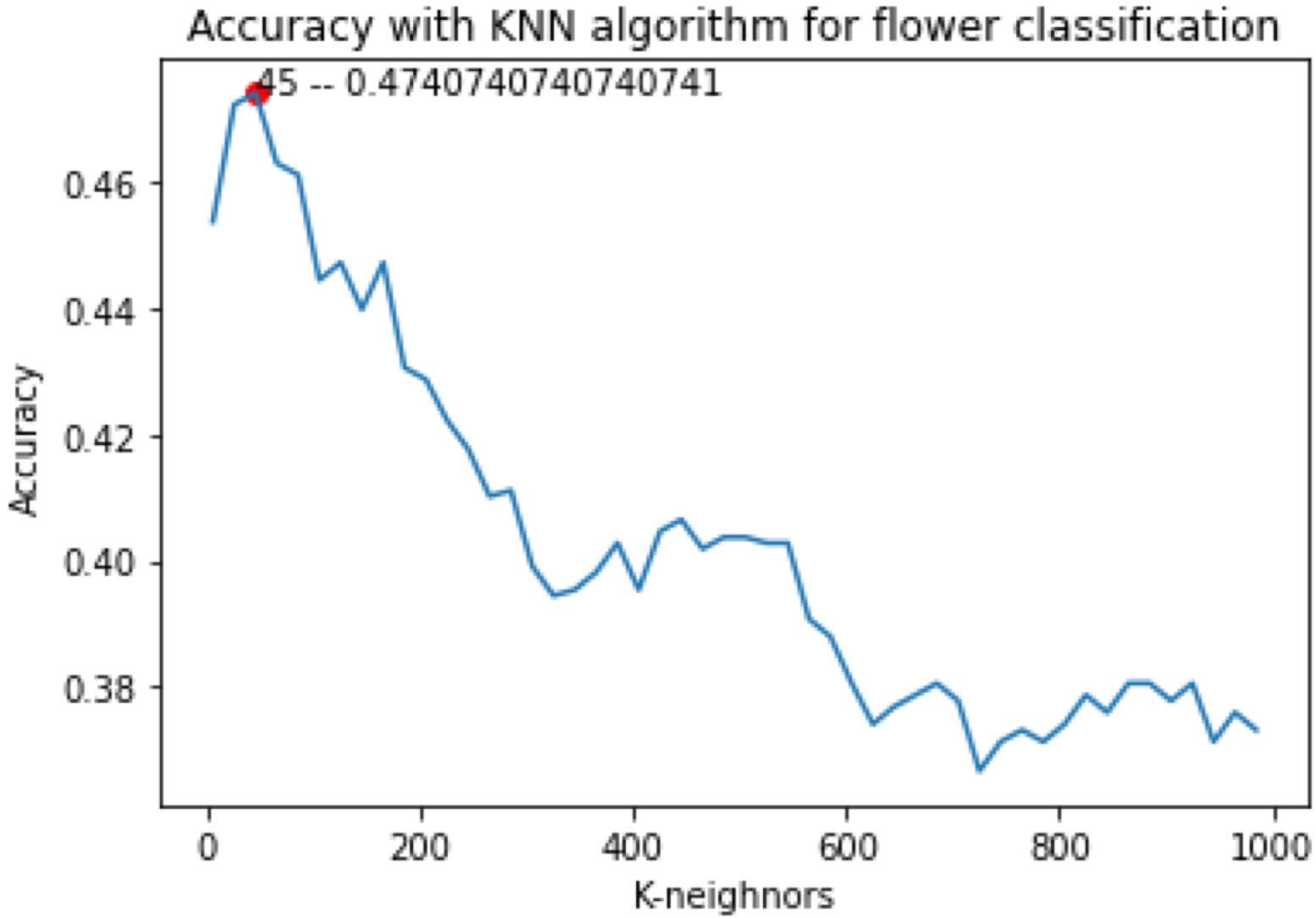


KNN classification with different K value

- KNeighborsClassifier (from scikit-learn)
  - **n_neighbors:** int, default=5
  - **weights:** {‘uniform’, ‘distance’} or callable, default=’uniform’
  - **algorithm:** {‘auto’, ‘ball_tree’, ‘kd_tree’, ‘brute’}, default=’auto’
  - **p:** int, default=2 –> decide distance fomulation between data sample
- Odd number for K value from 5 to 1000 with 20 steps for each iteration

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

KNN classifier with various K value



Evaluation on KNN classifier

- Performance Metrics
  - Confusion matrix
  - Accuracy
  - Precision and Recall
  - f1_score

prediction labels	Daisy	Dandelion	Rose	Sunflower	Tulip
Daisy	91	70	29	4	12
Dandelion	47	149	26	12	15
Rose	20	36	92	2	38
Sunflower	30	40	20	94	17
Tulip	28	54	54	14	86

$$F_1 = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{TP}{TP + \frac{FN + FP}{2}}$$

References

- https://easychair.org/publications/open/jKTm
- https://matplotlib.org/stable/index.html
- https://seaborn.pydata.org/introduction.html
- https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html
- https://pandas.pydata.org/
- https://pypi.org/
- https://docs.opencv.org/4.x/d5/d26/tutorial_py_knn_understanding.html
- https://keras.io/examples/vision/image_classification_from_scratch/
- https://cs231n.github.io/convolutional-networks/
- https://www.kaggle.com/alxmamaev/flowers-recognition/metadata