DATS-6103: Data Mining Final Project Individual Report

Popular Attraction/Landmark Recognition Using Google Landmark Dataset

Introduction:

Image Processing, Image Recognition and Image Classification has become a buzzing topic in the field of Engineering and Computer Science. With increasing use of social applications, trend of capturing pictures alongside various popular landmark has always been in fashion. Our focus on this project has been to recognize a given popular landmark irrespective of different aspect ratio or the illumination of the presented image. Google-Landmarks-v2 (September 2019), a dataset released by Google is used as source of data in the project. In order to maintain uniformity, images in the dataset are resized to a fixed ratio. Histogram of Oriented Gradients (HOG) classifier is then used for feature extraction which were later stored as array. We then ran various models such as K-Nearest Neighbors, Logistic Regression, Decision Trees, Linear Support Vector Machine, Non-Linear Support Vector Machine, Naïve Bayes and Random Forest. Random Forest worked best for our given dataset with an accuracy of 68%.

The project was divided among three members as follows –

Sharmin Nerius Kantharia – Data Preprocessing and GUI

Saurav Mainali – Feature Extraction and GUI

Gayathri Chandrasekaran – Modelling

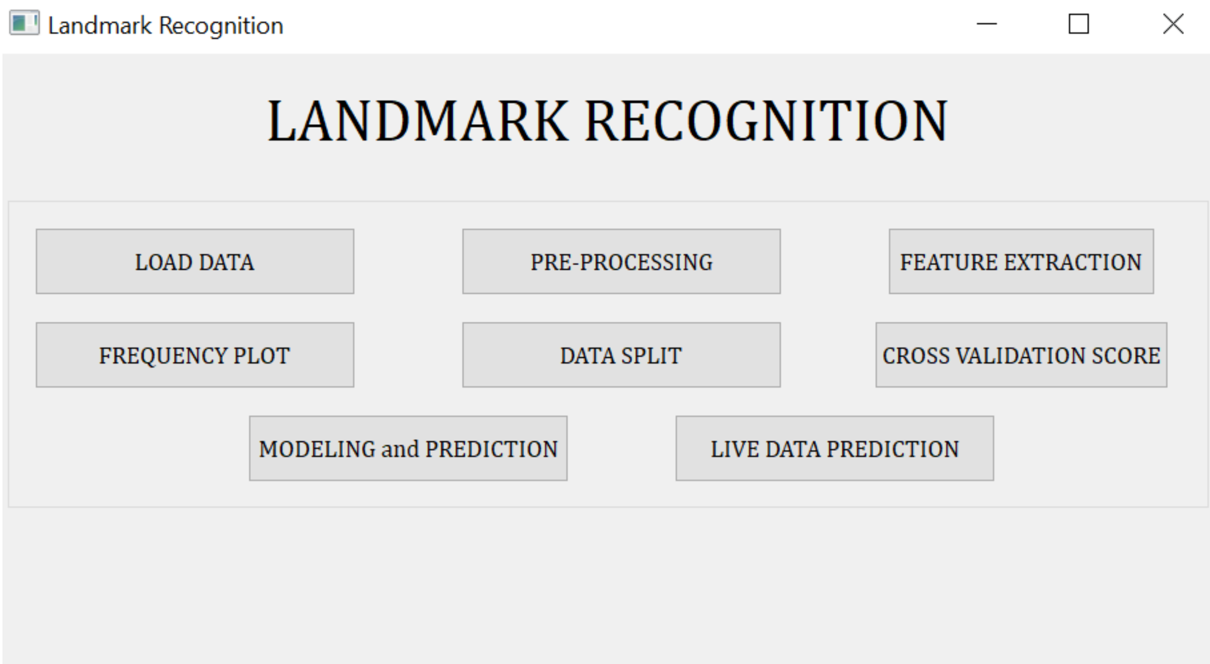
Feature Extraction

In order to get important features from the images HOG descriptor was used. Image was subdivided many cells. For every pixel in the image the HOG classifier calculates the gradients in both the horizontal and the vertical direction. Then magnitude and orientation of each gradients are measured and stored. Based upon this histogram is created and distributed over bins. Histograms for each subdivided cell is calculated which gives the features representing in that cell. Thereby important high-level information is generated. For this project, HOG classifier from OpenCV package was used.

Magnitude: $\sqrt{[(G_x)^2+(G_y)^2]}$

Orientation: $\Phi = \text{atan}(Gy / Gx)$

Graphical User Interface



For the GUI, main menu was created including multiple buttons. The buttons were named Load Data, Pre-Processing, Features Extraction, Frequency Plot, Data Split, Cross Validation Score, Modelling and Prediction and Live data and Prediction. Once the user interface was created using Qt designer, it was converted into .py file using the command prompt.

Results:

After the features were extracted we ran various model and the results are presented below.

EDA: Top 10 Sampled Landmark Details

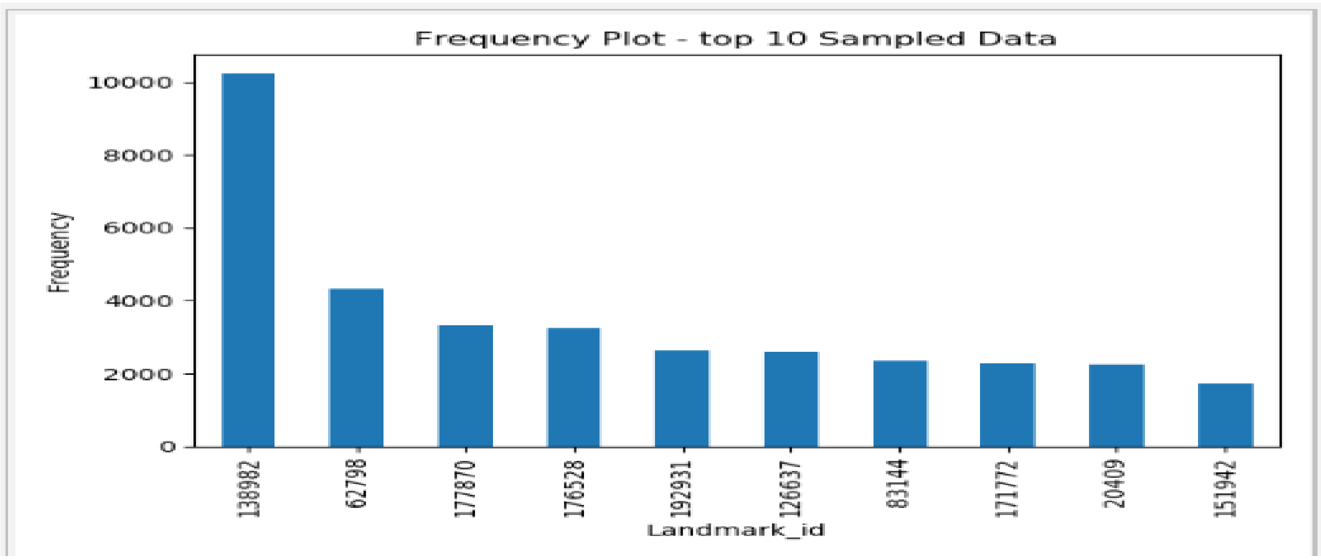


Fig 1: Frequency Plot – top 10 Sampled Data

The graph shows that out dataset is highly imbalanced. Landmark id – '138982' has highest annotated images.

CONFUSION MATRIX:

Confusion matrix for different model are described below.

Model Evaluation Metrics - Logistic Regression

Confusion Matrix	126637.0	138982.0	151942.0	171772.0	176528.0	177870.0
126637.0	378	77	20	37	14	1
138982.0	17	922	39	3	31	28
151942.0	19	119	201	38	32	19
171772.0	29	19	35	430	23	2
176528.0	19	15	4	16	2954	0
177870.0	17	190	43	9	14	41
192931.0	6	72	6	50	56	3
20409.0	70	267	20	27	27	7
62798.0	42	38	62	130	57	12
83144.0	39	170	70	44	34	17

	192931.0	20409.0	62798.0	83144.0
126637.0	6	34	58	32
138982.0	35	136	22	50
151942.0	16	103	97	62
171772.0	34	13	153	36
176528.0	11	11	34	10
177870.0	18	108	35	42
192931.0	374	43	38	14
20409.0	42	447	37	23
62798.0	44	51	513	85
83144.0	33	42	139	211

Model Evaluation Metrics - Random Forest

Confusion Matrix	126637.0	138982.0	151942.0	171772.0	176528.0	177870
126637.0	396	77	19	29	5	1
138982.0	2	1140	12	1	4	14
151942.0	30	71	254	40	3	16
171772.0	15	22	7	483	6	0
176528.0	4	22	3	14	2942	1
177870.0	26	187	65	8	2	89
192931.0	0	21	1	20	13	1
20409.0	20	291	22	30	8	16
62798.0	43	70	66	122	13	13
83144.0	20	152	79	41	6	14

	192931.0	20409.0	62798.0	83144.0
126637.0	6	43	39	42
138982.0	23	41	9	37
151942.0	28	64	106	94
171772.0	37	15	154	35
176528.0	18	18	45	7
177870.0	17	54	29	40
192931.0	565	3	17	21
20409.0	41	492	16	31
62798.0	38	49	513	107
83144.0	35	38	135	279

Fig 3: Confusion Matrix – Logistic regression & Random Forest

Model Evaluation Metrics - SVM Linear Model

Confusion Matrix	126637.0	138982.0	151942.0	171772.0	176528.0	177870.
126637.0	16	9	13	35	53	23
138982.0	50	211	27	25	351	46
151942.0	7	35	21	51	41	17
171772.0	10	13	65	191	43	30
176528.0	29	14	82	969	1237	4
177870.0	7	67	21	25	50	26
192931.0	2	4	15	175	9	3
20409.0	15	81	18	50	129	40
62798.0	26	29	70	101	103	34
83144.0	12	50	17	73	92	31

	192931.0	20409.0	62798.0	83144.0
126637.0	3	233	159	113
138982.0	22	212	154	185
151942.0	3	236	162	133
171772.0	4	120	211	87
176528.0	21	436	156	126
177870.0	6	137	86	92
192931.0	7	357	64	26
20409.0	5	451	84	94
62798.0	9	177	305	180
83144.0	4	168	186	166

Model Evaluation Metrics - SVM Non Linear Model

Confusion Matrix	126637.0	138982.0	151942.0	171772.0	176528.0	177870.0
126637.0	0	0	0	0	0	0
138982.0	0	157	0	0	0	0
151942.0	0	0	0	0	0	0
171772.0	0	0	0	0	0	0
176528.0	0	0	0	0	0	0
177870.0	0	0	0	0	0	0
192931.0	0	0	0	0	0	0
20409.0	0	0	0	0	0	0
62798.0	0	0	0	0	0	0
83144.0	0	0	0	0	0	0

	192931.0	20409.0	62798.0	83144.0
126637.0	0	0	0	657
138982.0	0	0	0	1126
151942.0	0	0	0	706
171772.0	0	0	0	774
176528.0	0	0	0	3074
177870.0	0	0	0	517
192931.0	0	0	0	662
20409.0	0	3	0	964
62798.0	0	0	0	1034
83144.0	0	0	0	799

Fig 4: Confusion Matrix – SVM Model

Model Evaluation Metrics - Decision Tree

Confusion Matrix	126637.0	138982.0	151942.0	171772.0	176528.0	177870.0
126637.0	283	42	44	47	10	28
138982.0	32	789	38	14	10	89
151942.0	52	50	154	46	18	68
171772.0	53	14	55	284	23	22
176528.0	18	27	21	34	2859	15
177870.0	24	113	58	15	6	96
192931.0	9	22	23	40	20	17
20409.0	59	183	62	50	15	79
62798.0	60	54	102	145	34	64
83144.0	38	80	90	63	23	79

	192931.0	20409.0	62798.0	83144.0
126637.0	10	68	79	46
138982.0	19	153	50	89
151942.0	23	68	128	99
171772.0	36	31	172	84
176528.0	16	19	40	25
177870.0	12	75	59	59
192931.0	426	33	42	30
20409.0	42	350	59	68
62798.0	44	60	313	158
83144.0	41	63	138	184

Model Evaluation Metrics - KNN

Confusion Matrix	126637.0	138982.0	151942.0	171772.0	176528.0	177870
126637.0	406	59	56	28	10	9
138982.0	45	922	29	3	5	70
151942.0	127	84	203	40	38	35
171772.0	80	21	34	407	98	3
176528.0	39	27	29	39	2863	3
177870.0	27	190	66	4	12	86
192931.0	10	42	7	57	89	4
20409.0	67	300	53	16	20	69
62798.0	155	43	135	157	97	33
83144.0	130	108	102	50	66	45

	192931.0	20409.0	62798.0	83144.0
126637.0	7	24	29	29
138982.0	40	121	10	38
151942.0	13	36	67	63
171772.0	22	11	67	31
176528.0	15	12	32	15
177870.0	30	57	11	34
192931.0	404	25	12	12
20409.0	76	328	12	26
62798.0	21	31	274	88
83144.0	25	37	75	161

Fig 5: Confusion Matrix – Decision Tree & KNN

Model Evaluation Metrics - Naive Bayes

Confusion Matrix

	126637.0	138982.0	151942.0	171772.0	176528.0	177870.0
126637.0	301	52	29	14	185	7
138982.0	113	695	104	18	47	76
151942.0	165	102	118	58	97	36
171772.0	111	32	22	229	195	15
176528.0	37	31	5	17	2911	3
177870.0	110	175	39	19	56	39
192931.0	36	10	22	26	291	2
20409.0	282	207	44	39	183	24
62798.0	228	134	86	122	142	46
83144.0	138	148	96	76	70	58

	192931.0	20409.0	62798.0	83144.0
126637.0	14	14	14	27
138982.0	29	79	15	107
151942.0	51	8	22	49
171772.0	81	4	46	39
176528.0	29	6	29	6
177870.0	24	13	7	35
192931.0	247	6	11	11
20409.0	72	69	20	27
62798.0	71	22	81	102
83144.0	46	14	33	120

Model Evaluation Metrics - Ensemble

Accuracy Score : 67.5833094624272

Confusion Matrix

	126637.0	138982.0	151942.0	171772.0	176528.0	177870.
126637.0	393	77	29	34	5	2
138982.0	8	1123	21	2	3	19
151942.0	31	77	257	40	3	14
171772.0	19	23	11	486	4	0
176528.0	5	23	6	15	2941	0
177870.0	23	189	68	8	4	88
192931.0	0	19	4	22	13	2
20409.0	23	277	28	30	9	16
62798.0	47	68	78	138	14	8
83144.0	21	150	81	44	9	16

	192931.0	20409.0	62798.0	83144.0
126637.0	6	40	37	34
138982.0	17	42	16	32
151942.0	29	64	109	82
171772.0	34	14	146	37
176528.0	16	20	40	8
177870.0	21	50	30	36
192931.0	567	2	17	16
20409.0	46	484	25	29
62798.0	39	41	474	127
83144.0	42	37	134	265

Fig 6: Confusion Matrix – Naïve Bayes & Ensemble Model

We can see landmark id – 177870,151942 has more misclassified records. SVM has low performance in our dataset. Landmark id - 177870,151942 has similar features with other classes which increases number of incorrect predictions.

CLASSIFICATION REPORT:

Model Evaluation Metrics - Logistic Regression

Classification Report

	precision	recall	f1-score	support
20409.0	0.59	0.58	0.58	657
62798.0	0.49	0.72	0.58	1283
83144.0	0.40	0.28	0.33	706
126637.0	0.55	0.56	0.55	774
138982.0	0.91	0.96	0.94	3074
151942.0	0.32	0.08	0.13	517
171772.0	0.61	0.56	0.59	662
176528.0	0.45	0.46	0.46	967
177870.0	0.46	0.50	0.48	1034
192931.0	0.37	0.26	0.31	799

Model Evaluation Metrics - Random Forest

Classification Report

	precision	recall	f1-score	support
20409.0	0.71	0.60	0.65	657
62798.0	0.56	0.89	0.68	1283
83144.0	0.48	0.36	0.41	706
126637.0	0.61	0.62	0.62	774
138982.0	0.98	0.96	0.97	3074
151942.0	0.54	0.17	0.26	517
171772.0	0.70	0.85	0.77	662
176528.0	0.60	0.51	0.55	967
177870.0	0.48	0.50	0.49	1034
192931.0	0.40	0.35	0.37	799

Model Evaluation Metrics - SVM Linear Model

Classification Report

	precision	recall	f1-score	support
20409.0	0.09	0.02	0.04	657
62798.0	0.41	0.16	0.23	1283
83144.0	0.06	0.03	0.04	706
126637.0	0.11	0.25	0.15	774
138982.0	0.59	0.40	0.48	3074
151942.0	0.10	0.05	0.07	517
171772.0	0.08	0.01	0.02	662
176528.0	0.18	0.47	0.26	967
177870.0	0.19	0.29	0.23	1034
192931.0	0.14	0.21	0.17	799

Model Evaluation Metrics - SVM Non Linear Model

Classification Report

	precision	recall	f1-score	support
20409.0	0.00	0.00	0.00	657
62798.0	1.00	0.12	0.22	1283
83144.0	0.00	0.00	0.00	706
126637.0	0.00	0.00	0.00	774
138982.0	0.00	0.00	0.00	3074
151942.0	0.00	0.00	0.00	517
171772.0	0.00	0.00	0.00	662
176528.0	1.00	0.00	0.01	967
177870.0	0.00	0.00	0.00	1034
192931.0	0.08	1.00	0.14	799

Model Evaluation Metrics - Decision Tree

Classification Report

	precision	recall	f1-score	support
20409.0	0.45	0.43	0.44	657
62798.0	0.57	0.61	0.59	1283
83144.0	0.24	0.22	0.23	706
126637.0	0.38	0.37	0.38	774
138982.0	0.95	0.93	0.94	3074
151942.0	0.17	0.19	0.18	517
171772.0	0.64	0.64	0.64	662
176528.0	0.38	0.36	0.37	967
177870.0	0.29	0.30	0.30	1034
192931.0	0.22	0.23	0.22	799

Model Evaluation Metrics - KNN

Classification Report

	precision	recall	f1-score	support
20409.0	0.37	0.62	0.47	657
62798.0	0.51	0.72	0.60	1283
83144.0	0.28	0.29	0.29	706
126637.0	0.51	0.53	0.52	774
138982.0	0.87	0.93	0.90	3074
151942.0	0.24	0.17	0.20	517
171772.0	0.62	0.61	0.61	662
176528.0	0.48	0.34	0.40	967
177870.0	0.47	0.26	0.34	1034
192931.0	0.32	0.20	0.25	799

Fig 7: Classification report

Model Evaluation Metrics - Naive Bayes

Classification Report

	precision	recall	f1-score	support
20409.0	0.20	0.46	0.28	657
62798.0	0.44	0.54	0.48	1283
83144.0	0.21	0.17	0.19	706
126637.0	0.37	0.30	0.33	774
138982.0	0.70	0.95	0.80	3074
151942.0	0.13	0.08	0.09	517
171772.0	0.37	0.37	0.37	662
176528.0	0.29	0.07	0.11	967
177870.0	0.29	0.08	0.12	1034
192931.0	0.23	0.15	0.18	799

Model Evaluation Metrics - Ensemble

Classification Report

	precision	recall	f1-score	support
20409.0	0.69	0.60	0.64	657
62798.0	0.55	0.88	0.68	1283
83144.0	0.44	0.36	0.40	706
126637.0	0.59	0.63	0.61	774
138982.0	0.98	0.96	0.97	3074
151942.0	0.53	0.17	0.26	517
171772.0	0.69	0.86	0.77	662
176528.0	0.61	0.50	0.55	967
177870.0	0.46	0.46	0.46	1034
192931.0	0.40	0.33	0.36	799

Fig 7a: Classification report

F1 scores for other algorithms are good. Landmark id – 138982 is classified properly most and Landmark id – 192931 is misclassified quiet often. Due to imbalance in dataset and multiclass environment, we are not able to conclude much from these results for other classes.

ACCURACY & COHEN KAPPA SCORE:

	Model	Accuracy_Score	Cohen Kappa Score
0	Logistic Regression	61.787453	0.548704
1	Random Forest	68.299437	0.627768
2	SVM_Linear	25.121742	0.141461
3	SVM Non Linear	9.156880	0.015796
4	Decision Tree	54.788504	0.471286
5	KNN	57.805786	0.503012
6	Naive Bayes	45.927623	0.351123
7	Ensemble(Hard voting)	67.583309	0.619471

Fig 8: Accuracy & Cohen Kappa Score

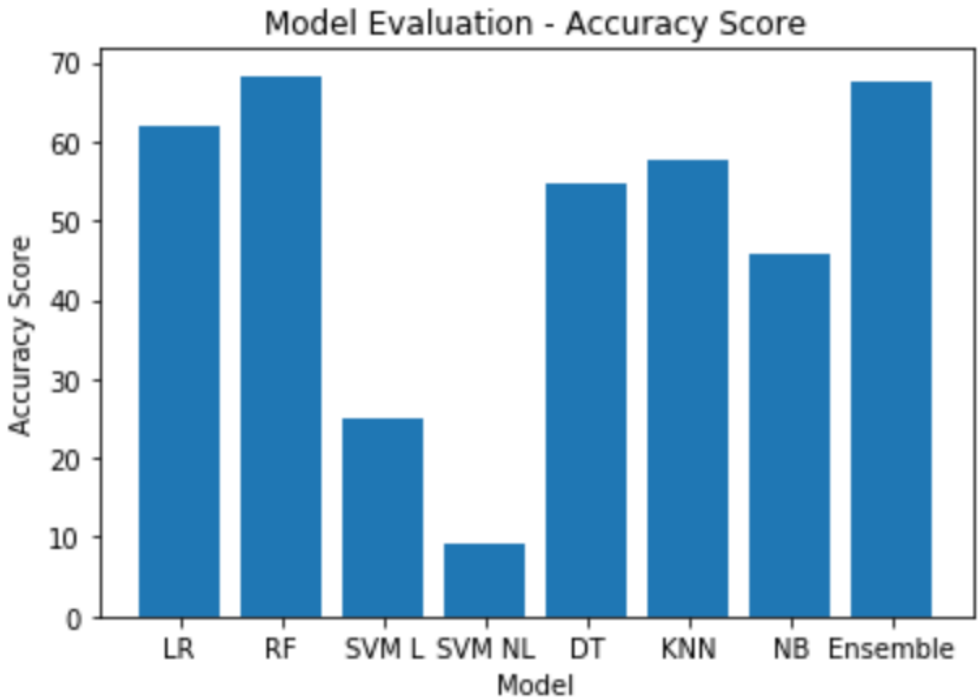


Fig 9: Accuracy Score for various Models

From Accuracy Score and Kappa Score, we can conclude Random forest gives better accuracy rate and kappa value also falls under Fair agreement region, followed by Ensemble and Logistic Regression. SVM model has lowest accuracy and kappa score is least, it doesn't suit for given dataset.

CROSS VALIDATION SCORE:

Model CV Score

	Classifier_Name	Cross_validation_Score
0	LR	0.617887
1	KNN	0.582703
2	Decision Tree	0.552306
3	RF	0.675163
4	NB	0.463691
5	SVM Non Linear	0.090619
6	SVM Linear	0.227345

Fig10: Cross Validation Score

The cross-validation score suggest us that, Random forest and logistic regression have better score and SVM models has the least.

Conclusion

Hence after we conducted test with all the models, we can conclude that Random Forest works best for the given set of data which gives 68% accuracy. Logistic Regression is the second best. SVM does not produce accurate result in our dataset. This is because our dataset is highly imbalanced and hence optimum decision boundary cannot be created.

Percentage of work from internet – I have used hog classifier form open cv algorithm tutorial for feature extraction which is 10 lines of code. Total percentage of code from internet is 17%

Reference Materials:

1. Announcing Google-Landmarks-v2: An Improved Dataset for Landmark Recognition & Retrieval (2019, September),
   Retrieved from: https://ai.googleblog.com/2019/05/announcing-google-landmarks-v2-improved.html
2. The Common Visual Data Foundation(2019, September), Google Landmarks Dataset v2,
   Retrieved from: https://www.kaggle.com/c/landmark-recognition-2019
3. Y. Li, D. J. Crandal and D. P. Huttenlocher, Landmark Classification in Large-scale Image Collections,
   Retrieved from: https://www.cs.cornell.edu/~yuli/papers/landmark.pdf
4. A. Crudge, W. Thomas and K. Zhu, Landmark Recognition Using Machine Learning,
   Retrieved from:
   http://cs229.stanford.edu/proj2014/Andrew%20Crudge,%20Will%20Thomas,%20Kaiyuan%20Zhu,%20Landmark%20Recognition%20Using%20Machine%20Learning.pdf
5. Y. Takeuchi, P. Gros, M. Hebert and K. Ikeuchi, Visual Learning for Landmark Recognition,
   Retrieved from: https://www.cs.cmu.edu/~takeuchi/iuw97/iuw97.html
6. https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/