MASK RCNN
Application Of DataScience Course Work Project
GROUP Q

Muhammad Atif Khan 45697647
Ajay Babu 45578192
Aisha Hasan Shah 45920842

November 15, 2019

1 Brief description of the source paper, and justification

Recently image processing has gained great charm and some terrific work. Amongst various fields of research and experimentation object detection in an image and also providing a pixel level segmentation(object instance segmentation) is challenging and Mask RCNN helps in solving this issue

Mask RCNN is conceptually simple and general framework for object instance segmentation.This network is an extended work on faster RCNN and is based on a CNN (Convolutional Neural Network) backbone structure. The paper claims "Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps. Moreover, Mask R-CNN is easy to generalize to other tasks, e.g., allowing us to estimate human poses in the same framework."

The researchers carried out comparisons with existing state of the art along with comprehensive ablation experiments. The original work was trained and tested on COCO dataset. The chosen metric for evaluation in the paper [1] was AP (averaged over IoU[2]thresholds) at different scales as depicted in Figure 1. The paper asserts that the system outperformed all the existing single models on the following three tasks; 1. instance segmentation 2. bounding-box object detection 3. person key point detection

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MNC [7]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [21] +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [21] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN	ResNeXt-101-FPN	**37.1**	**60.0**	**39.4**	**16.9**	**39.9**	**53.5**

Table 1. **Instance segmentation** *mask* AP on COCO `test-dev`. MNC [7] and FCIS [21] are the winners of the COCO 2015 and 2016 segmentation challenges, respectively. Without bells and whistles, Mask R-CNN outperforms the more complex FCIS+++, which includes multi-scale train/test, horizontal flip test, and OHEM [30]. All entries are *single-model* results.

Figure 1: Metrics as per the research paper.

Then paper was presented in 2017 in International Conference on Computer Vision (ICCV) and was published by IEEE in its journal IEEE Xplore and the citation stats on IEEE Xplore website are 724 citations in papers. Google scholar statistics provides a count of 3920 citations by 2019.

[1] Link for the Orginal paper https://arxiv.org/abs/1703.06870

[2] $IoU = AreaofOverlap/AreaofUnion$

2 Description of original dataset

The dataset used to train MaskRCNN is from COCO dataset [3].COCO stands for Common Objects in Context and the goal of this dataset is to advance the state-of-the-art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding. This is achieved by gathering images of complex everyday scenes containing common objects in their natural context.

COCO training dataset is unique and it contains 330K images collection with more than 200K labelled images. It has 1.5 million object instances along with super pixel stuff segmentation. Prominent features of this dataset include about 80 object categories and 91 stuff categories to create a better performing object detection models that can be further improved by the 5 captions per image feature offered by COCO.

3 Replication of original work

Mask RCNN gained much recognition and as evident from the citations for the paper many people tried to replicate the original work. The team followed the "matterport" [4] repository that implemented Mask RCNN for object detection and instance segmentation which is implemented on Keras[5] and TensorFlow[6].Our work is uploaded in Github(https://github.com/aishahassanshah/Data-Science/)

Initial task was to set a working environment which would handle the size of the COCO dataset and new data. The work was initiated by setting up the "EC2" instance on AWS. We had constant issues just to setup an envirnoment, so we went ahead with "Amazon SageMaker". Amazon sage maker is a fully managed machine learning service which has an integrated JupyterLab notebook [7] instance and sets up the secure and scalable environment adjusted to a specific workflow.

Trouble was faced resolving the dependencies, partially because the Github repository is not well maintained there are 1101 existing issues. The code was not compatible on new version of Python (3.7) as most of the inbuilt functions

[3]Link for the Original Dataset http://cocodataset.org/download

[4]https://github.com/matterport/

[5]Keras is TensorFlow's high-level API for building and training deep learning models.

[6]TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library and is also used for machine learning applications such as neural networks. (Source Wikipedia)

[7]JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning. JupyterLab is extensible and modular

were deprecated. When we referred to the ReadMe file present in the repository, it came to our knowledge that it was not updated. Other than that, we found certain required packages missing from the repository like "Pycocotools", a package which was required to run the MaskRCNN. Furthermore, Python was not the only thing that got updated, TensorFlow also released their latest version. Tensorflow (2.0) was incompatible with the MaskRCNN scripts so we had to revert to the older version on TensorFlow version 1.13.1.

In order to run the model on COCO dataset, we had to download the Validation dataset [8] which contained 5k images,which when compared to other split of the dataset was less and the Jupyter Instance was able to handle this size of the dataset.We loaded the pretrained weights of trained coco dataset and was successful in replicating the original work and obtain similar results.On the images, we could see the Bounding Box,Mask and also label for the objects on the image.The original code did not have any function which would save the output image from model,so we used a function which would take the coordinates of the Masks,Bounding Box of the image and later plot different color for each object.

Figure 2: Result of COCO Dataset

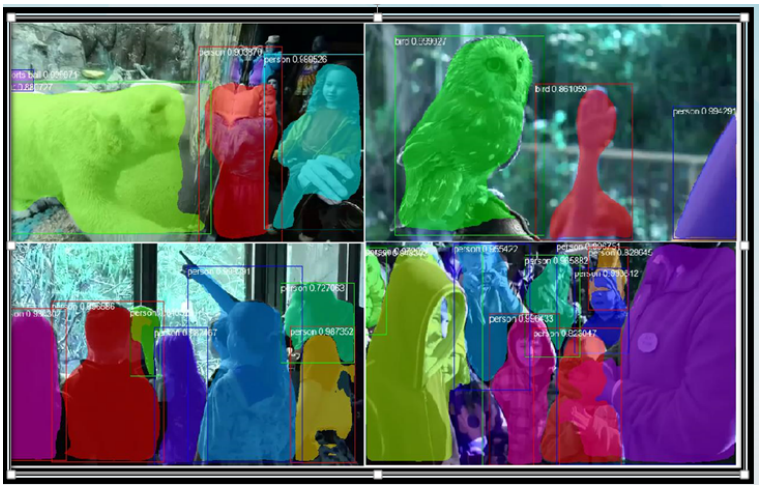
4 Construction of new data

For the new data we focused on 3 different ways. Since our data was images so one idea was to extract images from videos. We conducted two surveys to provide two measure's of agreement one with calculate Cohen's Kappa and other one providing descriptive statistics

[8] http://images.cocodataset.org/zips/val2017.zip

4.1 Videos

We recorded few videos and also downloaded few of the online video [9] .The reason for selecting a video is because we would contain lot many frames (images) and also many more objects based on scenario .To break the videos to frame we relied on Opencv [10] library and proivde these images to the MaskRCNN .



4.2 Google Open Images ver4(OIDV4)

The next choice was OIDV4 -open image Dataset V4 [11] (provided by google). This dataset is approximately 9 million images that have been annotated with image-level labels and object bounding boxes. The images are very diverse and often contain complex scenes with several objects (8.4 per image on average) and the dataset is annotated with image-level labels spanning thousands of classes.

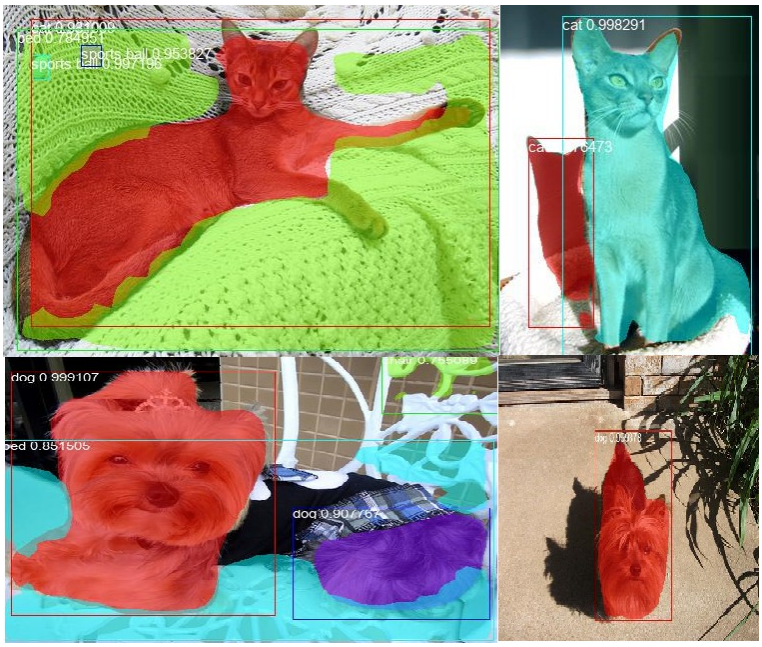
[9] https://www.youtube.com/watch?v=mX5TjFS4xsk
[10] https://opencv.org/
[11] https://opensource.google/projects/open-images-dataset

Again, the task was to figure out a way to extract a subset, a kind of class of images. To our rescue there was the OIDv4_ToolKit from the treasure house-GitHub. This amazing toolkit enabled the download of images from OID v4 seamlessly. The toolkit cloned the 600 classes along with their validations, annotations and bounding boxes.

4.3 Kaggle

For our third dataset [12] we approached to use the Kaggle Dataset, the reason being the popularity of the resource to download clean and structured datasets. We decided to go ahead with Cats and Dogs Breeds Classification Oxford Dataset which contains 37 category per dataset with roughly 200 images for each class. The dataset was recently updated by Dr. Avicenna 7 months ago (Version 1). The categories used were • Abyssinian (cat data set) • yorkshire_terrier (dog data set)

[12] https://www.kaggle.com/zippyz/cats-and-dogs-breeds-classification-oxford-datase



5 Results on new data

We decided to go ahead with human evaluation rather than classification metrics which was mentioned in the paper[13].The reason for this decision was because we had to annotate all the images.Each image contained multiple objects which would take lot of time for all three dataset.We conducted a survey using Google Forms and uploaded 8 images of each mentioned above dataset. We got a response of 45 people for each images. We will provide a descripitve statistics of this survey .We conducted one more survey with 50 images for each custom dataset and conducted a survey with two people .We later evaluated based on Cohen's Kappa to show how well the agreement is for two annotators on the custom dataset. In the survey we have provided a rating system where 1 is considered poor classification , 2 is considered moderate classification and 3 is considered as perfect classification In the Figure 3 we can see descriptive statistics of the 40 people survey, we can see that the average results are above 2.3 which suggests that model is doing well for classification on multiple objects as well.In the histogram provided below for each custom dataset we can see it follows a normal distribution where the majority of the rating is for 2(moderate/neutral) . In the Figure 4 we can see comparisons with other datasets, we can see that results obtained on OpenCV is poor, this maybe because on the video which were recorded on the phone contained objects which were not present on the COCO dataset and made wrong label classification.

[13]https://docs.google.com/a/students.mq.edu.au/forms/d/1yCK1QX6O3xOpzh65ZM-0upoHtSuWzTRk5Tol3Ik1MzM/edit?usp=drivesdk

We also calculated Cohen's kappa coefficient is a statistic that is used to measure inter-rater reliability (and also Intra-rater reliability) for qualitative (categorical) items. The calculation for Cohen's Kappa is done based on another survey which contains 50 each for each custom dataset and provided to two annotators

In Figure 13 we can see Cohen's Kappa value is 0.54 for COCO data set,which shows moderate agreement between two Annotators (A and B).

In Figure 14 we can see Cohen's Kappa value is 0.40 for openCV data set,which shows fair agreement between two Annotators (A and B).

In Figure 15 we can see Cohen's Kappa value 0.44 for Kaggle data set,which shows moderate agreement between two Annotators (A and B).

In Figure 16 we can see Cohen's Kappa value 0.30 for OIDV4 data set,which shows fair agreement between two Annotators (A and B).

Statistics

Variable	Mean	SE Mean	StDev	Mode	N for Mode
COCO Image 1	2.6182	0.0710	0.5267	3	35
COCO Image 2	2.2909	0.0960	0.7116	3	24
Kaggle Image 1	2.382	0.102	0.757	3	30
Kaggle Image 2	2.4000	0.0957	0.7097	3	29
OpenImages Image 2	2.4545	0.0929	0.6890	3	31
OpenImages Image 1	2.6182	0.0756	0.5608	3	36
Opencv Image 1	2.7455	0.0697	0.5170	3	43
Opencv Image 2	2.7636	0.0634	0.4700	3	43

Figure 3: Descriptive Statistics of the Data sets

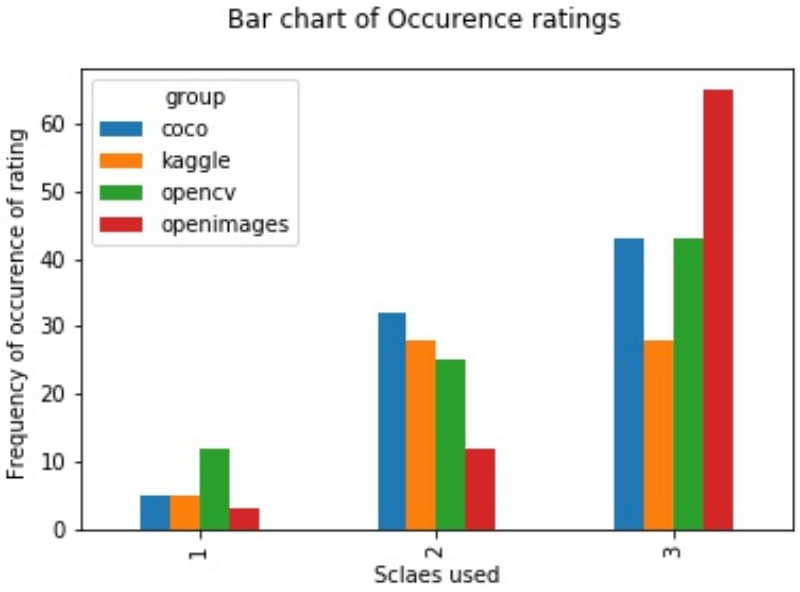


Figure 4: Relative Results of the Survey

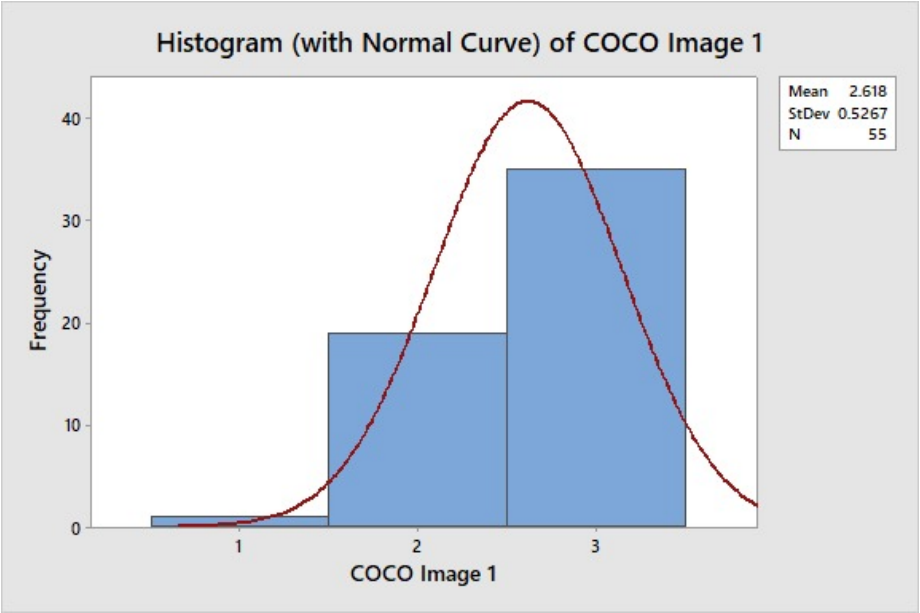


Figure 5: COCO image 1 Survey Result

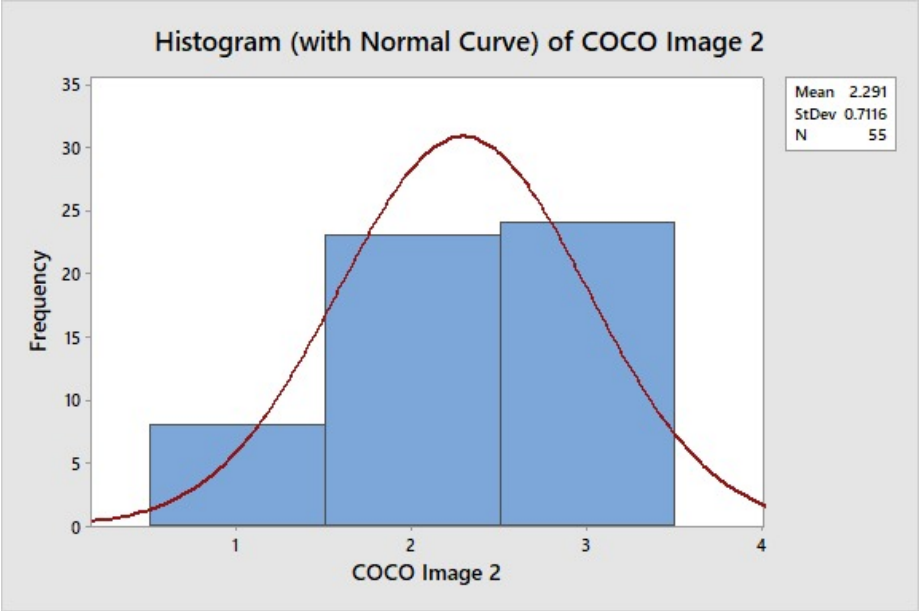


Figure 6: COCO image 2 Survey Result

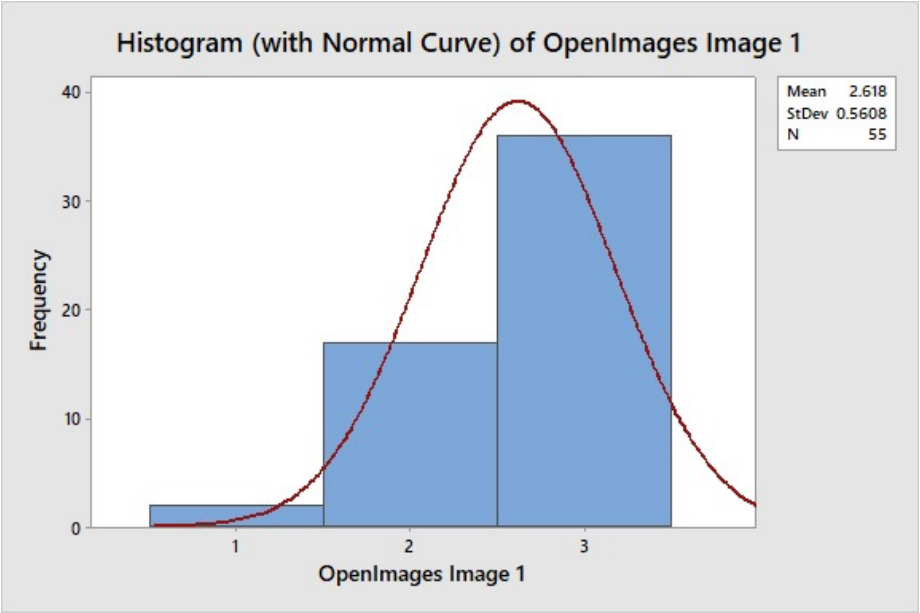


Figure 7: OpenImages image 1 Survey Result

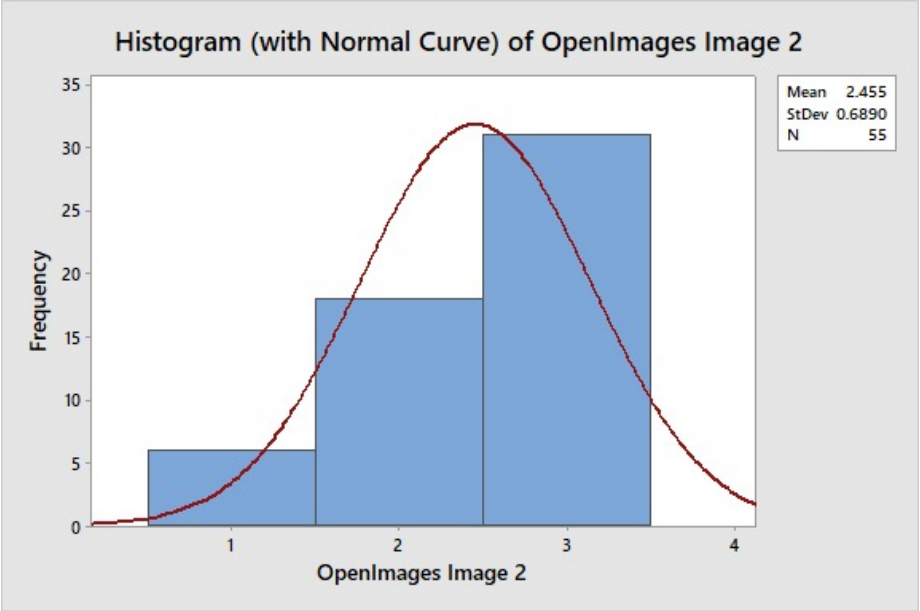


Figure 8: OpenImages image 2 Survey Result

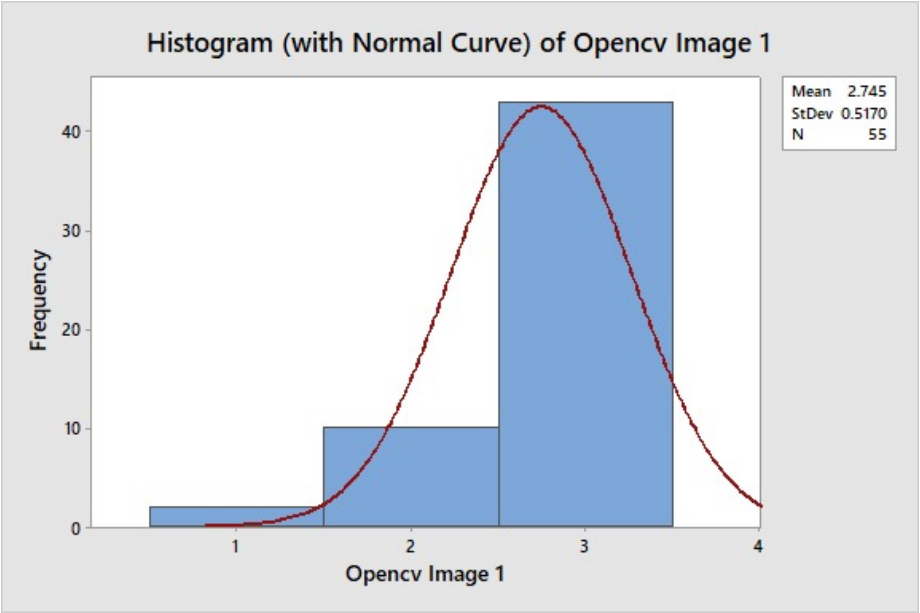


Figure 9: OpenCV image 1 Survey Result

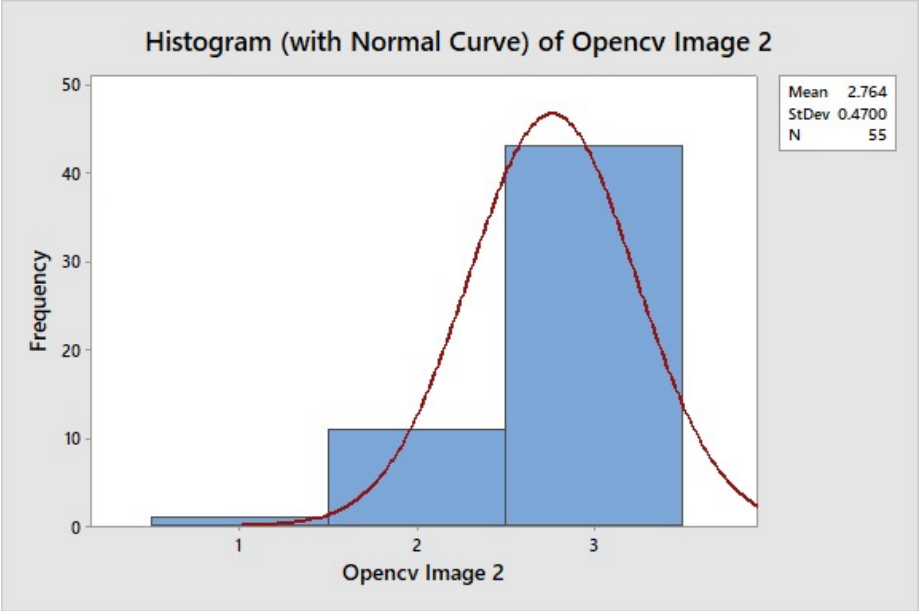


Figure 10: OpenCV image 2 Survey Result

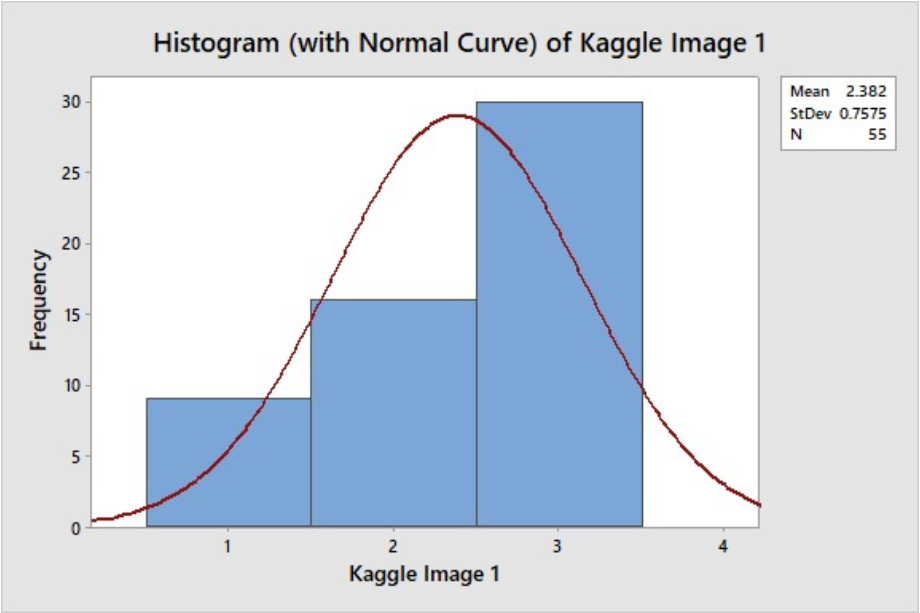


Figure 11: Kaggle image 1 Survey Result

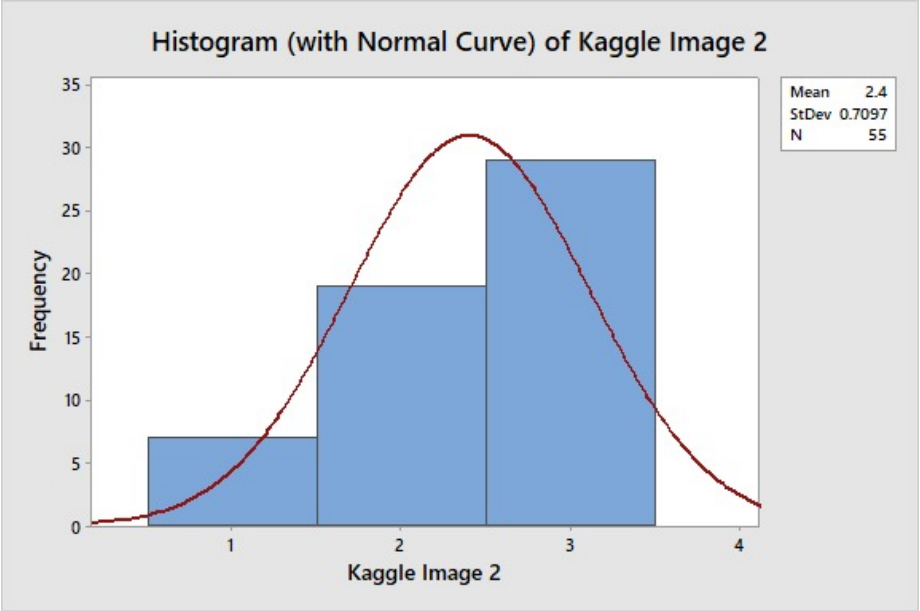


Figure 12: Kaggle image 2 Survey Result

Negative=1						
Neutral=2						
Positive=3			Annotator B			
		Negative=1	Neutral=2	Positive=3	Sum of rows	
	Negative=1	1	2	1	4	0.080
Annotator A	Neutral=2	3	8	2	13	0.260
	Positive=3	2	10	21	33	0.660
Sum of columns		6	20	24	50	
		0.120	0.400	0.480		

Cohen's Kappa k=Ao-Ae/1-Ae
where Ao is the observed Agreement which is here sum of diagonal of agreement of both Annotators A and B Ao 0.6

There are genrally 0.6 percentage of agreement exist between two Annotators on 50 images of COCO dataset.

Ae is expected agreement by chance between two Annotators A and B

Ao	0.74
Ae	0.4304
k	0.543539326

The Cohen's Kappa value is 0.5435 which is moderate agreement between two Annotator (A AND B) on 50 images of COCO data set

Figure 13: Cohen's Kappa measurement for COCO data set

			Annotator B			
		Negative=1	Neutral=2	Positive=3	Sum of rows	
	Negative=1	9	7	2	18	0.36
Annotator A	Neutral=2	3	10	4	17	0.34
	Positive=3	2	2	11	15	0.3
Sum of columns		14	19	17	50	
		0.28	0.38	0.34		

Cohen's Kappa k=Ao-Ae/1-Ae
where Ao is the observed Agreement which is here sum of diagonal of agreement of both Annotators Ao 0.6

There are genrally 0.6 percentage of agreement exist between two Annotators on 50 images of openCV

Ae is expected agreement by chance between two Annotat A and B

Ao	0.6
Ae	0.332
K	0.401197605

The Cohen's Kappa value is 0.40119 which is fair agreement between two Annotator (A AND B) on 50 images of OpenCV data set

Figure 14: Cohen's Kappa measurement for Opencv data set

		Annotator B				
		Negative=1	Neutral=2	Positive=3	Sum of rows	
Annotator A	Negative=1	1	1	2	4	0.08
	Neutral=2	0	2	1	3	0.06
	Positive=3	1	2	40	43	0.86
Sum of columns		2	5	43	50	
		0.04	0.1	0.86		

Cohen's Kappa k=Ao-Ae/1-Ae

where Ao is the observed Agreement which is here sum of diagonal of agreement of both Annotators Ao 0.86

There are genrally 0.86 percentage of agreement exist between two Annotators on 50 images of Kaggle dataset.

Ae is expected agreement by chance between two Annotators A and B

Ao	0.86
Ae	0.7488
K	0.442675159

The Cohen's Kappa value is 0.4426 which is moderate agreement between two Annotator (A AND B) on 50 images of Kaggle data set

Figure 15: Cohen's Kappa measurement for Kaggle data set

		Annotator B				
		Negative=1	Neutral=2	Positive=3	Sum of rows	
Annotator A	Negative=1	1	0	1	2	0.04
	Neutral=2	0	1	3	4	0.08
	Positive=3	2	1	41	44	0.88
Sum of columns		3	2	45	50	
		0.06	0.04	0.9		

Cohen's Kappa k=Ao-Ae/1-Ae

where Ao is the observed Agreement which is here sum of diagonal of agreement of both Annotators Ao 0.86

There are genrally 0.86 percentage of agreement exist between two Annotators on 50 images of OIDV4 dataset.

Ae is expected agreement by chance between t A and B

Ao	0.86
Ae	0.7976
K	0.308300395

The Cohen's Kappa value is 0.3083 which is fair agreement between two Annotator (A AND B) on 50 images of OIV4 data set

Figure 16: Cohen's Kappa measurement for OIDV4 data set