

Enhancing Object Clarity In Single Channel Night Vision
Images
Using Deep Reinforcement Learning

by

Adil Hossain

16201065

Mohammad Elham Robbani

17101132

Md. Riaz Ul Haque Sazid

17101292

Sk. Shahiduzzaman Siam

17101289

Wasiu Abtahee

20101623

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January,2021

© 2020. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Adil Hossain
16201065




Mohammad Elham Robbani
17101132



Md. Riaz Ul Haque Sazid
17101292



Sk. Shahiduzzaman Siam
17101289



Wsiu Abtahee
20101623

Approval

The thesis/project titled “Enhancing Object Clarity In Single Channel Night Vision Images Using Deep Reinforcement Learning” submitted by

1. Adil Hossain (16201065)
2. Mohammad Elham Robbani (17101132)
3. Md. Riaz Ul Haque Sazid (17101292)
4. Sk. Shahiduzzaman Siam (17101289)
5. Wasiu Abtahee (20101623)

Of Fall, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 11, 2021.

Examining Committee:

Supervisor:
Dr. Amitabha Chakrabarty
(Member)



Dr. Amitabha Chakrabarty
Associate Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
Dr. Md. Golam Rabiul Alam
(Member)



Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
Mahbubul Alam Majumdar
(Chair)

Mahbubul Alam Majumdar
Professor and Dean, School of Data and Sciences
Department of Computer Science and Engineering
BRAC University

Ethics Statement

As we collected our primary data from surveillance areas of different companies, factories, corporate offices. We are respondent that all the identity of the footage will be kept anonymous and this data will only be used for research purposes.

Abstract

There are a lot of novel approaches to image processing using Machine learning and classical image processing. But most of them take a huge dataset[like machine learning] or they are slow and inefficient[like plain image processing]. Keeping this in context it was always a center of attraction to solve the problem of denoising and clarity enhancing in night vision images. As night vision images are like an asset sometimes, most of the CCTV footage is considered in this. Because if we think simply a complete footage of a CCTV has 50 percent of the recording in day time and 50 percent in night time. And just like that there is 50 percent chance of capturing any event at night. Now if we consider a crime scene which has to be extracted from the cctv footage at night time there is a huge probability of the footage to contain noise , distortion and clarity compromisation. In these scenarios identity extraction is difficult. But along the progress of computation we have image processing and machine learning to develop and filter these images but we talked about the disadvantages before. To train a big dataset for extracting 1 footage is not feasible. So, we are considering a new novel approach which is also considered as state-of-the-art approach of using AI to filter a noisy night time single channel image and enhancing clarity and retain identity in it. In this work we will be facing limited resources and gradually developing those images by training an intelligent agent based on reward bias. So that after training the agent a limited resource it can predict future pixels based on it's reward bias. Our approach on processing the images will consider deep Q learning and using a convolutional network based on Q learning. Our Aim will be to retain most of the information dealing with these limited resources using a reinforcement learning based approach built with the above stated structure.

Keywords: limited dataset, Deep Q learning, AI , intelligent agent , reinforcement learning , Deep Q network, single channel images, night cctv footage

Dedication

Our research is dedicated to individuals those who lost their valuable life in this pandemic and those who lost their dear ones in this pandemic. This work is also little tribute to those frontiers who were fighting against novel coronavirus risking their own lives.

Acknowledgement

Above all else, we would like to thank Almighty for his great help which caused us to proceed with our thesis work without facing any kind of major difficulties in this pandemic situation. Secondly, we would like to thank our supervisor for tolerating our errors and sending continuous feedback to proceed with our work. We are also grateful to our parents who supported us in various ways to complete our research work.

Table of Contents

Declaration	i
Approval	ii
Ethics Statement	iv
Abstract	v
Dedication	vi
Acknowledgment	vii
Table of Contents	viii
List of Figures	x
List of Tables	xi
Nomenclature	xi
1 Introduction	1
1.1 Introduction	1
1.2 Problem Statement	1
1.3 Aim of Study	2
1.4 Research Methodology	2
1.5 Thesis Outline	3
2 Related Work	5
3 Data Collection and Added Feature	8
3.1 Data Collection	8
3.2 Data Relevancy	10
3.3 Added Feature	12
3.3.1 Variance of Laplacian	12
3.3.2 Salt And Pepper Noise	15
3.4 Feature Analysis	19
4 Model Selection and Results Analysis	22
4.1 Q-Mapping	22
4.2 Asynchronous Advantage Actor Critic(A3C)	23
4.3 Pixel Reinforcement Learning	24

4.4	Edge Detection for blur reduction	26
4.5	Result and Analysis	30
4.5.1	Reward Test results	30
4.6	PSNR Test results	36
4.7	Result analysis and Result summary	42
5	Conclusion and Future Work	43
	Bibliography	47

List of Figures

1.1	shows our workflow and steps of our thesis outline.	4
3.1	: Stages of data collection and preprocessing	9
3.2	: Graphical representation of PSNR of our initial data	11
3.3	: Example of a noisy image data representing the dataset. [12] . . .	12
3.4	: Example of a high laplacian variance image representing blurry image in dataset	14
3.5	: Example of a high PSNR count added with salt and pepper image representing noisy image in dataset.[31]	17
3.6	: Graph representation of Laplacian variance of the dataset	19
3.7	: Graph representation of PSNR count of the dataset	20
3.8	: Process overview of adding features	21
4.1	Training process for the proposed model, updating the prediction towards all goals at once.[23]	23
4.2	Network architecture of the fully convolutional A3C. The numbers in the table denote the filter size, dilation factor, and output channels respectively. [24]	25
4.3	Q-Learning framework architecture. [8]	27
4.4	The proposed automatic parameterization for the Sobel edge detector. [8]	28
4.5	Reward Test result of 0-200 episode	30
4.6	Reward Test result of 200-400 episode	30
4.7	Reward Test result of 400-600 episode	31
4.8	Reward Test result of 600-800 episode	31
4.9	Reward Test result of 800-900 and 900-1000 episode	32
4.10	Reward Test result of 1000-3000 episode	33
4.11	Superposition of Reward Test results of 0-1000 episode	34
4.12	Superposition of Reward Test results of 0-3000 episode	35
4.13	PSNR Test results of 0-200 episodes trained model output image . . .	36
4.14	PSNR Test results of 200-400 episodes trained model output image .	37
4.15	PSNR Test results of 400-600 episodes trained model output image .	38
4.16	PSNR Test results of 600-800 episodes trained model output image .	38
4.17	PSNR Test results of 800-1000 episodes trained model output image	39
4.18	PSNR Test results of 1000-3000 episodes trained model output image	39
4.19	ACCUMULATED PSNR VALUE 0-1000	40
4.20	ACCUMULATED PSNR VALUE 0-3000	40
4.21	Final Result	42

List of Tables

3.1	: Table showing the variance of the laplacian count of the dataset . . .	15
3.2	: Table showing the PSNR count of the salt and pepper noise added dataset	18

Chapter 1

Introduction

1.1 Introduction

Almost everyone takes pictures and videos with a purpose of sharing it on our social media. But nowadays pictures and videos are not only used for entertainment purposes but also as evidence, data and so on. So devices also vary depending on the purpose of the pictures and videos that are captured. For example, in case of professional photography people use DSLR, for medical purposes people use microscopic devices, for security purposes CC cameras etc. Capturing quality differs from device to device according to their uses. The same video recorder via 2 different devices such as CC camera and DSLR camera will never be the same as the devices have two different dedicated purposes. Here CC camera's footage has lower resolutions rather than DSLR footages. But CC cameras are used for security purposes which is more valuable for further investigations. Low resolution footage made it a tough job for investigators because of lower pixel footages, moreover things get more tougher when it is night time footage. As single channel images in CCTV footage things are hardly recognised so we thought we can take those single channel images of CC camera footage to a prominent level where people can easily identify those objects that they are desiring to see or investigate. So we looked at many studies related to this topic. One of them was "Waterloo Exploration Database: New Challenges for Image Quality Assessment Models". In their paper to assess the generalization ability and aid the broad application of Image Quality Assessment (IQA), they made a large scale database called Waterloo Exploration database having 4744 clear natural images and 94880 distorted images.[21] We focused on the outcomes of the footage with more clarity, details, less noise which will be discussed in details in various sections of our study.

1.2 Problem Statement

In the era of the 21st century still we can see the newspapers where some incidents could not be investigated properly due to lack of vivid footage. Almost every significant place has their own CC cameras but if any occurrence happened target objects can not be recognised for heavy noise footages, objects located at very long distance. As a result their target does not achieve for which they establish a camera there. Also there is a drastic change of footages during daylight and night time. Night footages are more noisy and less vivid due to less light and fog and other floating air

particles etc. So it is harder to find the target object from night footage. Basically night footages are grayscale. So desired object identification also gets tougher for not having RGB scale. However, this problem happens to almost every CC camera footage at night time. So we decided to work on this particular sector which may help in a broader ranger to get rid of this problem and to objectify target objects at night time as well.

1.3 Aim of Study

Analysing single channel night vision footage and enhancing vividity of footage to objectify the target object is the main goal of our thesis. We will be taking AI based approaches which will include deep reinforcement learning and Q learning based on reward bais methodes. Here we have discussed in detail about from which part of view we relate night vision CC camera footage to single channel image. How those footages are related to less clarity images and more noisy images. We also give rise to all those questionnaires which are connected to limitations of diminished pixel of images to pursue target object identification. Moreover our sumptuous dataset which hooked up to all real single channel images helps to outcome a more valorie image to objectify the target. These particular features more or less answer those questionnaires. Apart from this, the algorithm we used here suits perfect in the box to get desired output for which we are looking for. After running that model we noticed that the features we are looking for have come out with the output data. And this was we will be completing an implementation of AI in image processing

1.4 Research Methodology

Our main vision is to make single channel images more comprehensive where objects are complicated for bisecting in details. Keeping this vision in mind we started to collect data from different CC camera footages at night which are basically single channel images. We focused to get those data where objects can not be labelled properly due to lacking of details. We targeted open source CC camera footages but here were less footages which are at night vision. So we came up with building our own dataset. For our own dataset we collected many footage from different areas where they remain under surveillance all the time under CC cameras. We also used the Waterloo exploration database. To make our dataset usable for our model we pre process footage into single channel footage. Some of the footage was from daylight. So we converted those footage into grey scaled footage by using the `cvtColor` function in OpenCV. After that we made those footage into individual single frames. Those footages finally got ready to be our dataset. From those we selected a portion for our test dataset and another portion for our train dataset. We created a deep Q network based on reinforcement learning. Where a neural network which is approximate to the Q- value function where inputs are given as state and output is generated all the possible values of the Q- value function. As our main goal is to maximize the value function of Q. Which helps us to gain low pixelated images to our desired output image. Our implementation is based on a3c on python in ChainerRL library. Where learning begins from episode one to the last number of episodes. Results of each hundred number of episodes we saved as a model. End of

the training we saved the results of outcome reward as our testing model. Moreover, each model of hundred episodes tested according to our testing dataset. We also use here fully Convolutional Network (FCN). The benefits we got here by using FCN is that the shareable parameters and learn effectiveness of all agents. We will ensure data relainvency based peak signal to noise ratio (PSNR) for noise reductions and variance of laplacian variance to get rid of blurriness. Finally reinforcement learning with pixel-wise rewards (PixelIRL) takes us to get the results.

1.5 Thesis Outline

Our study keeps impacts on establishing a model for betterment of objectifying any kind of object no matter if that is a single channel image or normal image. Here our authors high aim is to work on a dataset on the context of our country which can be use in the furthestmost research work in this particular field of work to classify a whole new observation. The whole study light on the steps which were followed by researchers.

Firstly, the introduction part (Chapter 1) focuses on the motivation for which authors were on that particular problem for solving. Here mainly the focal objective and our work briefly talked here.

Secondly, in the literature review portion (Chapter 2) we talked about various papers from computer science background which also discussed almost similar topics. The main goal of background study was discovering the inadequacies of past investigations. Moreover, we have expressed our commitment and purposes for essential information assortment.

Thirdly, in the data collection section (Chapter 3) , we explained the way we collected the dataset and mixed Waterloo dataset for preparing test train splits. After data collection procedure there we did some data preprocessing. Then on the later part of this chapter we discussed the integrity of the dataset along with PSNR analysis and count so that we ensure the dataset is relevant. Feature selection discussed about the embedded feature of the dataset we will be considering for training and testing. We show the laplacian variance and PSNR analysis of the dataset which confirms the feature of our ground truth.

Fourthly , Model Selection (Chapter 4), includes the building procedure of the model we proposed. We showed how the Q learning model is gradually developed by embedding into convolutional network thus creating a multi step deep Q learning environment,

Furthermore in the result analysis section we mapped all the test output data and visualized to prove our proposed ground truth.

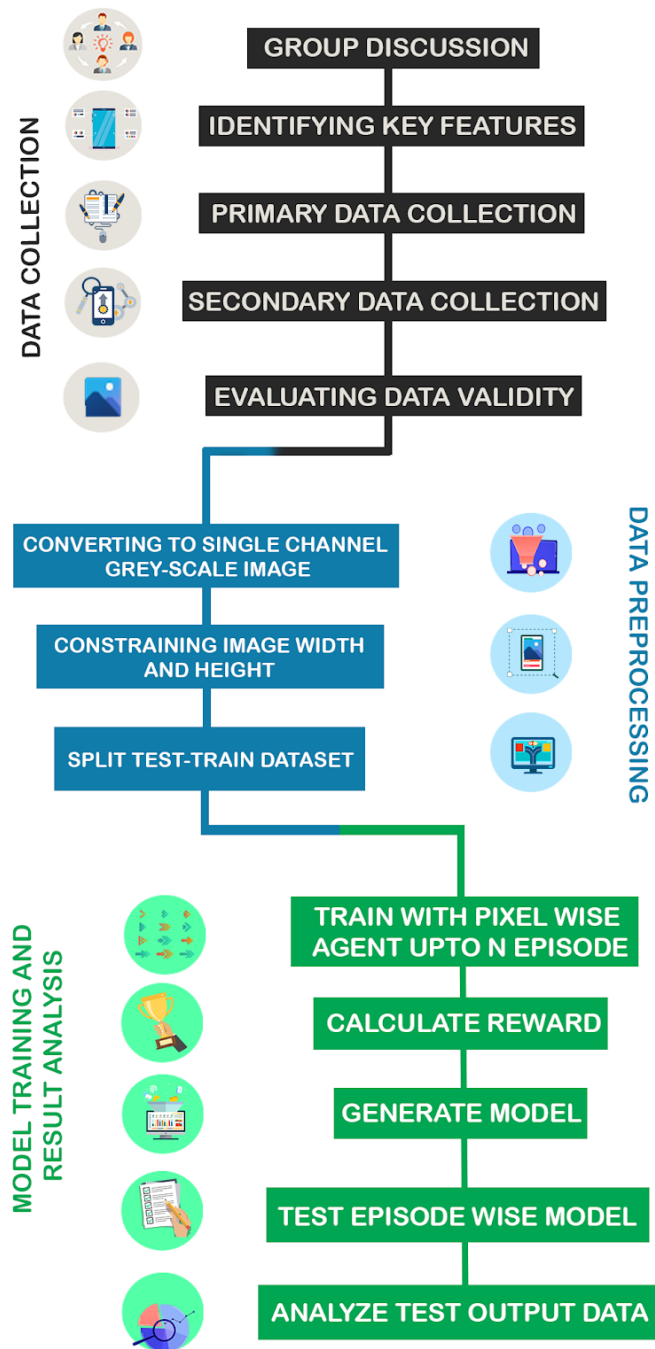


Figure 1.1: shows our workflow and steps of our thesis outline.

Figure 1.1 shows our workflow and steps of our thesis outline

Chapter 2

Related Work

Currently machine learning is one of the fundamental advancements utilized in settling different tasks such as picture division, feature detection, object acknowledgment and following. Modernized and complex systems such as robots are outfitted with visual sensors from which they learn the condition of the general environment by solving corresponding PC vision assignments or tasks and the solutions to these tasks utilized for settling on choices about conceivable future actions. Reinforcement learning is one of current AI technologies in which learning is helped out through association and interchange with the environment. Lately, Reinforcement learning has been utilized both for addressing such applied undertakings as handling and investigation of visual data, and for taking care of specific computer vision issues, for example extricating picture highlights, limiting items in scenes, and numerous others. [29]

Fundamentally, RL calculation causes the operator to pick activities under some particular strategies, which expand rewards gathered from a successive consequence of the activities. Reinforcement learning has empowered to manage a high dimensional state space with the goal that crude pictures can be considered as states. Be that as it may, utilizing crude pictures as state spaces need broad computational assets at times due to their high measurement, so there has been a lot of study utilizing additional calculations, for example, Convolutional Neural Network (CNN) to separate highlights from pictures. At the same time those algorithms decrease the computational load, it is still a very intricate process. [9]

With deep reinforcement learning(DRL) add the autonomous system to a higher level of understanding. The DQN or deep Q network establishes a reward biased platform where the algorithm developed a capability to predict the future steps or moves, thus without need of a classed dataset the system is able to explore and exploit unsupervised data and enhances decision making.[14]

Despite the limitation in the uses of deep reinforcement learning in terms of image processing, a new extension called pixel reinforcement learning had good results. An agent is set for a pixel so the number of agents are equal to the number of pixels and by taking an action the agent changes the value of the pixel. Additionally a proposal was given for a successful learning strategy for pixel reinforcement learning that fundamentally upgrades the exhibition by thinking about not just the future

conditions of the it's own pixel but also in addition to those of the neighboring pixels. The suggested technique can be implemented on some picture handling undertakings that need pixel-wise controls, where profound Reinforcement Learning has never been implemented before. [27]

Another form of reinforcement learning, Offline Reinforcement learning, alludes to the issue of taking in arrangements from a static dataset of environment interactions. It empowers broad use and re-utilization of historical datasets, while likewise reducing wellbeing concerns related with online investigation, consequently growing this real time a useability of RL. Most earlier works in offline RL have zeroed in on tasks with minimized state portrayals. Nonetheless, the capacity to gain straightforwardly from rich perception spaces like pictures is basic for true applications, for example, mechanical technology. [28]

A captivating Reinforcement Learning (RL) calculation, Asynchronous Advantage Actor Critic (A3C), is for a wide scope of undertakings. As an example, Atari games and bot control, where the agent learns strategies and worth capacity through experimentation cooperations with the climate until combining to an ideal approach. Dependability robustness are very important in RL; in any case, neural network can be powerless against clamor or noise from unanticipated sources and isn't probably going to withstand slight unsettling influences. We note that the agents created from gentle environment utilizing A3C can't deal with testing conditions. Gaining from antagonistic models, we proposed a calculation called Adversary Robust A3C (AR-A3C) to improve the agent's execution under boisterous conditions. In this calculation, an antagonistic agent is acquainted with the learning cycle to make it more hearty against unsettling influences or anomalies, consequently increasing its versatility to uproarious conditions. The two reproductions and certifiable analyses are done to represent the steadiness of the proposed calculation. The AR-A3C calculation beats A3C in both perfect and uproarious conditions. [26]

Ongoing exploration has demonstrated that map raw pixels from a solitary forward looking camera straightforwardly to directing orders are shockingly incredible. A research has been done where convolutional neural organization (CNN) to play the CarRacing-v0 utilizing impersonation learning in OpenAI Gym. The dataset is created by playing the game physically in Gym and utilized an information growth strategy to extend the dataset to multiple times bigger than before. Likewise, the genuine speed, four ABS sensors, controlling wheel position, and gyration for each picture and crafted a blended model by joining the sensor information and picture input. Subsequent to preparing, this model can consequently recognize the limits of street highlights and drive the robot like humans.[30]

Works like single-channel image blind reclamation by utilizing iterative principal component analysis (PCA) can be taken as reference. Beforehand it was seen that the iterative PCA approaches for dazzle rebuilding and demonstrated its prevalence over customary strategies. In any case, there are a few issues that are to be sorted out. An exact and automatic approach to decide the iteration number is one of the ways of trying to solve the issues. This research attempts to address this by

applying a visually impaired picture quality evaluation for automatic optimization of the iterative number. [10]

In this study the analyzing and synthesizing of frames, formalizing the BM3D picture displaying and utilizing these casings to create unique iterative deblurring algorithms. Two distinct plans of deblurring were shown: one given by minimization of the single target work and another dependent on the Nash equilibrium formulation of two target capacities. The last outcomes in an algorithm where the denoising and deblurring activities are decoupled. The intermingling of the created calculations is demonstrated. Simulation tests show that the decoupled calculation obtained from the Nash equilibrium formulation exhibits the best mathematical and visual outcomes and shows prevalence with deference over the best in class in the field, affirming a significant capability of BM3D-outlines as a high level picture displaying device. [7]

Asynchronous gradient descent for optimization of deep neural network controllers were used in a basic and lightweight system for Deep Reinforcement learning. The paper talks about a nonconcurrent variation of four standard reinforcement learning algorithms and demonstrates that actor-learners have a stabilizing effect permitting every one of the four techniques to effectively prepare neural organization regulators. [19]

Chapter 3

Data Collection and Added Feature

3.1 Data Collection

The inaccuracy and lack of the quality in the modernized night time recording equipment is something that is being worked on to improve the efficiency of the night time clips or images that are later used for various purposes. Our aim was to introduce a system that would ultimately improve the quality of the night time clips or images by a significant amount. Initially we searched about already existing systems that would help us get a better understanding of this which can be seen by our research in the literature review section. We wanted a system that would allow us to implement deep reinforcement learning on single channel night vision images to enhance the clarity of it.

The reason behind using reinforcement learning instead of the popular choice (machine learning) is because in reinforcement learning the system works on reward basis and due to this the system is able to work faster and needs less iterations than a system run by machine learning. We particularly targeted the night vision section because we wanted to improve the quality of the images we see nowadays.

Initially our group had countless discussion sessions as to how we are going to approach this mammoth like task, then after that discussions were done on which member will be looking out for which sections of this system. We planned out a workflow that would be good for the whole group and started working accordingly.

We used two kinds of data: Primary data and Secondary data. The data that is procured through first- hand research is known as Primary data. An example of this sort of data will be the data collected by us directly from cctv cameras from our building and neighbouring buildings. This was very tough to do as not everyone is cooperative and so we depended more on the Secondary data. The data type that was already collected by someone else. In this case we resorted to collecting the data from open sources present all over the internet. These datas are available to all for use for the general public.

The validity of our data was checked by PSNR method. PSNR stands for Peak signal-to-noise ratio which is a term indicating the ratio between max feasible power of a signal and the influence of corrupting noise that affects its representation's fidelity. The standard of reconstruction of lossy compression codecs is most generally calculated by PSNR.[3] Here using PSNR we compared the noise level of the data that we collected by setting a reference image to which all the data was compared with.

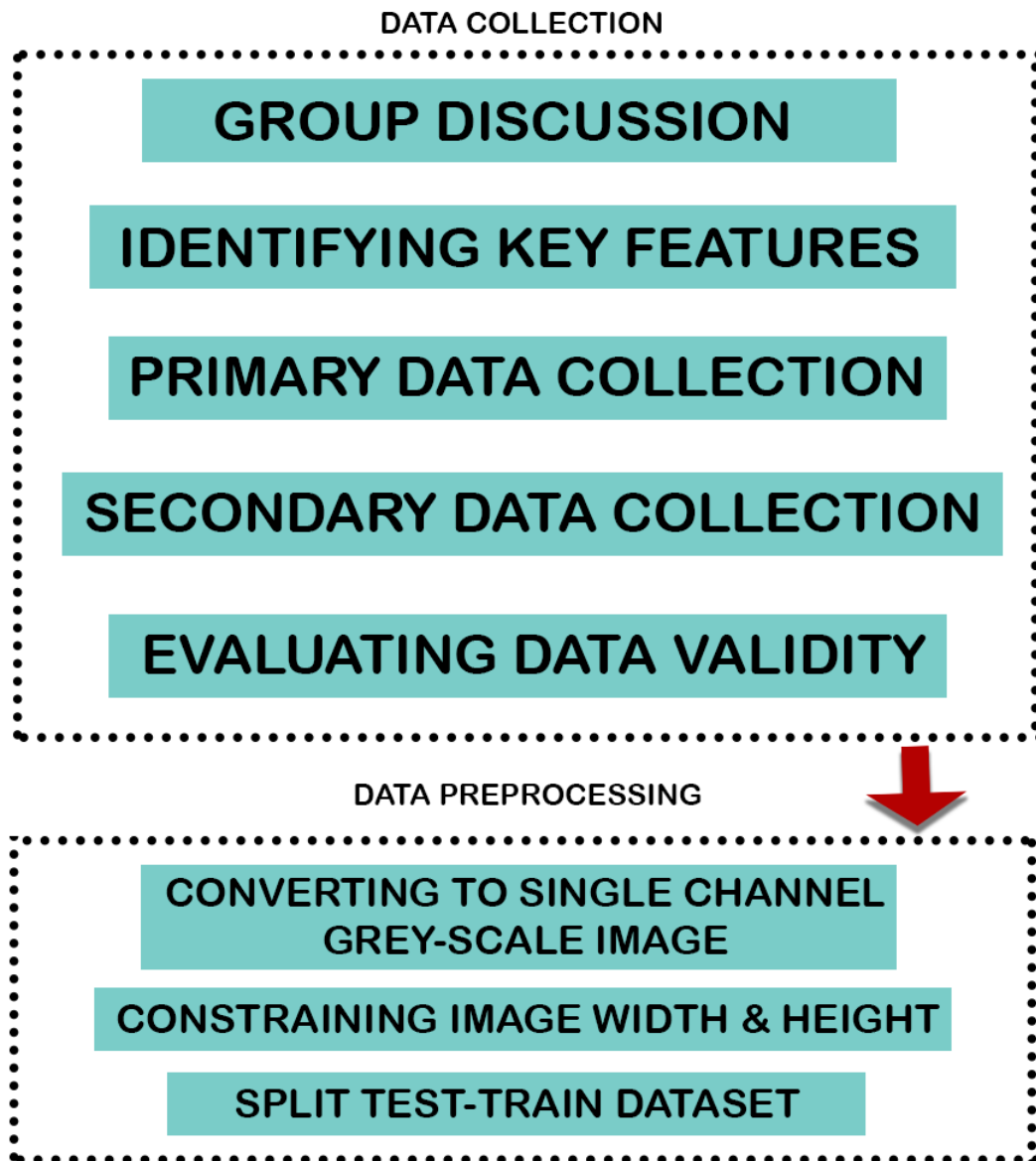


Figure 3.1: : Stages of data collection and preprocessing

In the data preprocessing part initially we turned all the images into single channeled grey scale images. The reason behind separating such pictures from some other kind of textured picture is that fewer data will be collected for every pixel. Indeed 'grey' texture is one in which the red, green and blue segments all have the same power in RGB space, thus it is simply salient to indicate a sole force as an incentive for every

pixel, instead of the three powers that are awaited to determine every pixel in a full shading picture. [4]

Frequently, the grayscale force is termed as a 8-digit whole number handing 256 potential numerous conceals of dark from dark to white. In the event that the levels are equally separated, at that point the distinction between progressive gray levels is altogether in a way that is better than the gray level settling intensity of the natural eye [4]

For our model we set the image canvas size to width 481 pixel and height 321 pixel. In general there is no problem with anysize of the image but for better efficiency and less storage cost we chose to keep the image size to 481 pixels width and 321 pixels height respectively keeping the image quality intact. This is all done after the conversion of the image to a single channel grey-scaled version of it.

According to our previous statement, we had 446 unique images with which we added a Waterloo dataset to increase the size of the train/test images. Waterloo Exploration database is a large scale database that holds around 4744 clean natural pictures and around 94880 distorted pictures, we used a chunk of data from this dataset. [21] After getting all the dataset in a place we split the dataset into 2 halves. One for testing and one for training. We trained our model with the majority portion of images from the dataset and then tested them against the minority of the pictures that we kept aside as they were better compared to the ones that our model trained with.

This concluded our data collection and data preprocessing part and in the next section we will be discussing the data relevancy of our data and how accurate is our data for our study.

3.2 Data Relevancy

Data relevancy basically means the regularity of data in the field of interest of the user. If a data does not match the field of use that it was procured for then that data is discarded as not relevant data.

The importance of data relevancy is emphasized in every field of education and work. We live in a world which is ruled by science, science is ruled by theories and these theories are proved by datas so if the datas are wrong then the theory is automatically incorrect which can lead to a very bad impact on that field of education/work. The legitimacy of a work or business is determined by the data or results that it provides and that result is solely dependent on the data.

To show that our data is relevant, we used PSNR. As stated before, PSNR stands for Peak Signal-to-noise ratio that indicates the ratio between max feasible power of a signal and the influence of corrupting noise that affects its representation's fidelity. [6]

PSNR is also defined by Mean Squared Error(MSE) which has a formula of:

$$MSE = \frac{1}{pq} \sum_{u=0}^p \sum_{v=0}^q [I(u, v) - K(u, v)]^2 \quad (3.1)$$

And PSNR is referred by:

$$\begin{aligned} PSNR &= 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \\ &= 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned} \quad (3.2)$$

It is usually used to evaluate the quality of rebuilding of lossy pressured codecs. The sign for this instance is the initial information, and the disturbance is the mistake due to the pressure or compression. When looking at pressure codecs, the PSNR count is an approximation of human impression of recreation quality.

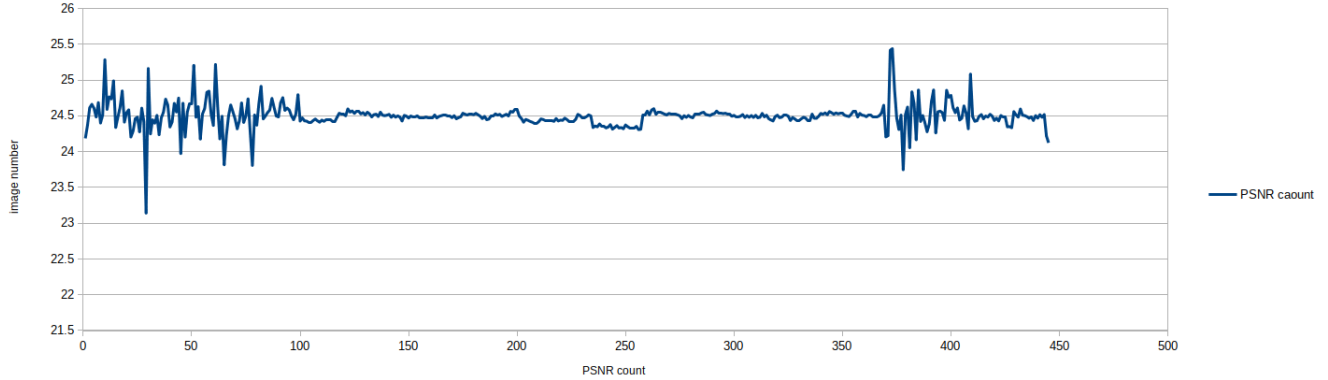


Figure 3.2: : Graphical representation of PSNR of our initial data

The above Figure 3.2 is the PNSR count of the images that we initially collected. It can be seen that the average value of the initial image ranges from 24.4 to 25. This count means that there is a decent amount of noise in the images. And this amount of noise is relevant for our model. Later on we will be increasing the noise count so that the effectiveness of our model can be demonstrated more easily.

Although the fact that a higher range of the PSNR value mostly demonstrates that the recreation is of higher caliber but it's not always the case. Being incredibly cautious with the range of legitimacy of this measurement, decisively legitimate when it is utilized to analyze results from the identical codec (or codec type) and similar content.[6]

An example of a noisy picture that we used in our dataset is given below:



Figure 3.3: : Example of a noisy image data representing the dataset. [12]

The PNSR count of this sample image which resembles most of the images that we collected to use as a dataset is 24.185236788799553. The value that we got here is a decent amount of noise which we will be increasing furthermore later on to see if our model can efficiently decrease the amount of noise a picture possesses.

Thus it can be concluded that the initial data that we have accumulated for our model is relevant and we can move forward with this study.

3.3 Added Feature

3.3.1 Variance of Laplacian

After our initial data collection and preprocessing which qualifies the Peak signal-to-noise ratio or PSNR then they need to be sorted according to the proposed features which will help us to reach the ground truth that we have claimed after we test our trained model. As we have discussed before our dataset is majorly emphasized on low quality images containing noise and blur scenarios. In the PSNR it has already qualified for the high noise level but another key feature which is our prime focus is blurred images. To make sure we are making our intelligent agent learn the right data we need to set some ground rules here too for ensuring data qualifications. For that we are going to analyze Variance of the Laplacian. [15]

The laplacian operator is expressed as :

$$\text{dst} = \Delta_{\text{src}} = \frac{\partial^2 \text{src}}{\partial a^2} + \frac{\partial^2 \text{src}}{\partial b^2} \quad (3.3)$$

We will be taking a pre processed a single channel image(grayscale converted) and convolve it in a 3x3 kernel:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (3.4)$$

After mapping the squared standard deviation of the response. If the variance is recorded within a threshold, at that point the picture is considered foggy or blur. The explanation for which it works is due to the influence of the Laplacian administrator, which is utilized to the second degree subordinate of the picture. As a result of this the laplacian variance generated can differentiate the quick intensity of the kernel changes, much or just like the Sobel and Scharr administrators. Similarly like this administrator the Laplacian variance is repeatedly used for edge discovery. The presumption here is that on the off chance that a picture contains high fluctuation at that point there's a wide spread of reactions, both edged and non-edged like agent of a ordinary focused picture. But in case there's quiet a low volume of variance change, at that point there's a minor spread of reactions, both edged and non-edges like, agent of a ordinary, in-focus picture. But in case there's exceptionally low variance, at that point With a modest spread of reactions, indicating there are exceptionally small edges within the picture. As we know, the more a picture is obscured, the less edges there are. [13]



Figure 3.4: : Example of a high laplacian variance image representing blurry image in dataset

Obviously the strategy here is setting the proper edge threshold which can be very space subordinate. As well moo of an edge and you'll erroneously stamp pictures as foggy when they are not. As well tall of a edge at that point pictures that are really hazy will not be stamped as hazy.

Thus by setting up the laplacian variance architecture we run the whole pre-processed dataset and analyze the data further in data visualization to confirm the relevance of the dataset , reliability and proving the ground truth of the work.

In the analysis we will take the combined dataset and run into the openCV Laplacian operator to extract the data and plot the data visualization for further analysis.[17] The opencv laplacian contains the similar algorithm built for our laplacian variance determination. After the iterative calculation the result Is obtained.An example visual of the obtained numerical data of the laplacian variance of the dataset has been shown for better understanding:

Image number	Laplacian Variance	Image Number	Laplacian Variance	Image Number	Laplacian Variance	Image Number	Laplacian Variance	Image Number	Laplacian Variance
1	4661.15347693492	51	3815.65921485119	101	5177.90826664824	350	5034.50713731642	400	4253.99132129478
2	4730.46741582918	52	4837.4633708914	102	5279.14965527721	351	5016.8468413107	401	4824.60505792452
3	4809.99033460943	53	4524.92592022676	103	5329.31490500962	352	4994.21277031348	402	4535.59787750895
4	4506.31061330414	54	5320.35484179618	104	5264.3348541009	353	5009.98801260595	403	4455.25124072625
5	4676.83396829495	55	4551.69398016071	105	5283.72371086793	354	4953.12514788691	404	4478.80186669954
6	5739.24568992949	56	4862.08405383635	106	5233.52273614684	355	4890.51003493092	405	4568.06497848779
7	5037.30499612154	57	4237.34587257834	107	5173.2763239751	356	4884.28310788713	406	4762.66676635094
8	4571.78138832896	58	4290.95907834613	108	5141.29597124012	357	4969.46949037591	407	4582.46697514538
9	4535.69101431704	59	4505.60827838004	109	5157.55490766904	358	4927.45840232682	408	4898.64823589233
10	4975.09175490227	60	4811.3754717867	110	5087.8509037332	359	4922.23138412191	409	3840.91585447383
11	4334.73285698975	61	3726.16033750242	111	5160.04797097903	360	4926.42841633033	410	5151.3169635315
12	4245.94671067377	62	4435.40957590915	112	5108.13562329246	361	4987.24068385844	411	5212.52431989892
13	4351.86888540357	63	5295.34981660537	113	5113.26562467663	362	4959.43576687418	412	5136.20036343952
14	4021.55085374795	64	4970.65363570661	114	5110.66783271219	363	4953.05345138993	413	4990.70875422721
15	4709.26235481379	65	4545.97405713464	115	5128.45408327224	364	4970.83054793459	414	5033.32461508671
16	4491.25276566303	66	5397.54179849766	116	5113.57686231213	365	4964.74205660572	415	5025.68280984232
17	4685.78521918216	67	4569.8017580762	117	5098.12232397448	366	4942.1757979438	416	4942.45735217502
18	4145.76908041853	68	4633.04848016449	118	5066.43501709997	367	4947.91502446939	417	4952.13525193864
19	4572.83187201909	69	4583.56137138501	119	5110.55230096661	368	4939.75294433915	418	4998.42444740506
20	4442.80584266813	70	4531.17721532116	120	5095.09191318096	369	4687.85339571198	419	4966.09891662978
21	5077.41554739035	71	5157.45383126961	121	5122.5455320884	370	4876.2679820495	420	4951.98159718083
22	4525.09432688397	72	4501.73642358989	122	5054.61841432743	371	4663.11695700424	421	4949.65586365874
23	4687.35420573497	73	4329.68681529864	123	5066.99890637799	372	3602.19952557754	422	4996.01530797718
24	4496.2518747566	74	4941.84603233918	124	5009.61784898977	373	3573.9108289	423	4968.77606153401
25	4672.58187265837	75	4524.08594972326	125	5056.9876562872	374	3933.52133564106	424	4974.46058505408
26	5332.81242173327	76	5327.79196510541	126	5027.03378497829	375	4508.61804630379	425	4952.86023165721
27	4488.00358829104	77	4542.11411380832	127	4999.7251892409	376	4857.66887691	426	4974.37645612214
28	4558.02242973711	78	6120.37107963461	128	5011.25365765689	377	4697.37648245913	427	4978.06681209833
29	6390.36696879235	79	4758.08196234169	129	4977.39300650733	378	5383.57308405176	428	4962.2072588182
30	3930.59901379273	80	5019.3212511961	130	4987.42308581613	379	4550.93395794069	429	5053.0424925392
31	4850.27375635645	81	4561.50157019583	131	4947.43083079714	380	4421.61495760849	430	5067.12451531285
32	4760.74333710422	82	4164.53802024487	132	4959.08317663888	381	4839.86755297734	431	5103.49764242406
33	4627.19865480173	83	4645.40447861812	133	4979.00949698372	382	4855.19307699916	432	5033.93927407054
34	4726.98329498483	84	4556.32082846108	134	4940.05927801916	383	4934.81699607662	433	5073.47399246443
35	5266.32487906324	85	4574.70969903908	135	4958.34937402502	384	5071.44294550206	434	5023.82867950541
36	4549.91068586364	86	5395.24265954042	136	4973.6972294675	385	5074.23655022849	435	5041.50434990718
37	4699.73293780039	87	4705.00782737812	137	4935.26650739522	386	4906.26259343191	436	5007.25250373163
38	4388.06457534619	88	4455.08552121298	138	4967.83886146515	387	4897.04097628457	437	5084.66832847869
39	4658.05082848508	89	4833.6662833264	139	4985.31080670268	388	4573.31117465511	438	5046.95369389397
40	4488.74143236265	90	4497.7338703292	140	4967.03104093363	389	5303.04105929219	439	5097.6778080397
41	4462.95435402396	91	4552.61015700333	141	4983.17456496922	390	5061.94552495713	440	4978.61610608774
42	4485.4611359523	92	4291.87044599613	142	5057.67898032007	391	4429.76434172862	441	4922.44916275916
43	4640.28268486892	93	4621.76844132383	143	5038.75669142992	392	5289.30726104965	442	4730.45662131429
44	4224.28208997542	94	4581.93226151364	144	5065.10426959828	393	5294.33796033935	443	4790.30484194024
45	4922.85304163777	95	4654.55624883889	145	5032.2209147736	394	4643.85012150547	444	4875.42484194024
46	4282.75345355088	96	4650.8849001641	146	5031.927824091	395	4616.22511369394	445	5023.93255326039
47	4775.6847031198	97	4548.89411780801	147	5065.39744834914	396	4686.26095492047		
48	4495.35814167245	98	4764.96492022721	148	4996.35517140027	397	4392.05955265259		
49	4221.63787142372	99	4071.77119835424	149	5018.32091347954	398	4286.25017365953		
50	4473.51438681633	100	4461.65410743703	150	5025.45240360691	399	4340.35859675666		

Table 3.1: : Table showing the variance of the laplacian count of the dataset

On the later part of this chapter we will visualize this obtained data and check the relevance of data for our proposed model

3.3.2 Salt And Pepper Noise

Salt and pepper noise is in some cases called impulse noise or spike noise or arbitrary noise. In salt and pepper noise, pixels in the picture are exceptionally diverse in color or concentrated not at all like their surrounding pixels. Salt and pepper corruption can be caused by sharp and sudden unsettling influence within the picture signal. Generally this sort of noise will as it were influence a little number of picture pixels. When seen, the picture contains dull and white specks, thus the term salt and pepper noise. Typical sources incorporate bits of tidy interior the camera and overheated or defective elements. An picture containing salt-and-pepper noise will have dark pixels in shining locales and This sort of noise can be caused by dead pixels,It known as incautious noise. It appearances is haphazardly scattered white or dark pixel over the picture.

Salt and pepper noise delivers a wide variety of processing and results in basic image distortion.It creates an effect of noise by accumulating dead pixels with a few noisy pixel, but they are very noisy. The impact is comparative to sprinkling white and dark dots—salt and pepper—on the image.

One case where salt and pepper noise emerges is in transmitting pictures over noisy digital joins. Let each pixel be quantized to A bits within the normal mold. The value of the pixel can be composed as

$$Y = \sum_{i=0}^{A-1} b_i 2^i \quad (3.5)$$

Accept the channel could be a twofold symmetric one with a hybrid likelihood of ε . At that point each bit is flipped with likelihood ε . Call the gotten esteem, X . At that point, accepting the bit flips are autonomous,

$$\Pr [|Y - X| = 2^i] = \varepsilon(1 - \varepsilon)^{A-1} \quad (3.6)$$

For $i = 0, 1, 2, \dots, m-1$. The MSE due to the foremost critical bit is $\varepsilon 4^{A-1}$ comparing to $e(4^{m-1}-1)/3$ for all the other bits combined. To say in other sentences, the commitment to the MSE from the foremost noteworthy bit is around three times that of all the other bits. The pixels whose most critical bits are changed will likely show up as dark or white dots. Salt and pepper clamor is an case of (exceptionally) overwhelming followed noise. A simple model is the taking after: Let $f(x,y)$ be the initial picture and $q(x,y)$ be the picture after it is modified by salt and pepper noise. [18]

$$\begin{aligned} \Pr[q = f] &= 1 - \gamma \\ \Pr[\mathbf{q} = \mathbf{MAX}] &= \gamma/2 \\ \Pr[\mathbf{q} = \mathbf{MIN}] &= \gamma/2 \end{aligned} \quad (3.7)$$

Here, MAX and MIN are the greatest along with the least picture values, separately. For 8 bit pictures, MIN = 0 and MAX = 255. The thought is that with likelihood 1 the pixels are unaltered; with likelihood γ the pixels are changed to the biggest or littlest values. The modified pixels see like dark and white specks sprinkled over the image. Figure 3.5 appears to show the impact of salt and pepper commotion. Around 15% of the pixels have been set to dark or white (85% are unaltered). Take note the sprinkling of the dark and white dabs. Salt and pepper noise is effectively expelled along different arrange measurement channels, particularly the center weighted middle and the LUM channel. [16]



Figure 3.5: : Example of a high PSNR count added with salt and pepper image representing noisy image in dataset.[31]

Salt and pepper noise, moreover known as information drop-out, can cause pointed and unexpected unsettling influences within the picture flag. The clamor thickness here is 0.01. The pictures mutilated by this commotion and the watermarks extricated from them are appeared in Figure 3.5 for standard pictures and restorative pictures. The objective measurements for the mutilated pictures are detailed in Table 3.2. Since this sort of clamor is particular (i.e., as it were a few pixels are supplanted, either by 255 or by 0), a recognizable watermark is extricated, but from the related table and figure, it is obvious that the impact of the commotion is unmistakable to the human eye and consequently the proposed framework is delicate to this particular clamor.

Keeping salt and pepper distortion as one of the major feature of our training model , the dataset was trimmed and after preprocessing we need to ensure the sp or any gaussian noise presence in the dataset so that we can prove the ground truth of removing the noise. Just like we discussed the PSNR in the dataset section we will be running the PSNR again to ensure gain of PSNR over the post stage of the dataset processing. The dataset will be going through the PSNR algorithm and the data is presented in the Table 3.2

Image number	PSNR	Image number	PSNR	Image number	PSNR	Image number	PSNR	Image number	PSNR	Image number	PSNR
1	10.1089709788	51	11.7245944964513	101	9.3479843958	301	11.67523789798	351	10.6205089611495	401	10.0100186295385
2	10.44574852843	52	9.8294451697312	102	9.858626858	302	11.6518194511	352	10.7023826257261	402	10.000507512732
3	10.92364607815	53	11.0891889882238	103	7.7420258521	303	11.65898846364	353	10.7529559502613	403	10.0143053872478
4	7.659530734305	54	11.5011115572776	104	10.856762443	304	11.66050375873	354	10.8913456625041	404	9.9669557019331
5	9.303376554391	55	7.10860902672471	105	10.946686098	305	10.59988351768	355	10.8786044119762	405	10.2077242694653
6	11.58351681616	56	10.8058858186804	106	9.5866334921	306	10.54584090362	356	10.879223833594	406	10.2347050144051
7	9.734407220468	57	7.96394651334175	107	9.5274126644	307	10.52652085424	357	10.886287488797	407	10.278193819275
8	9.13698333503	58	10.789130516208	108	10.183289152	308	10.48439233259	358	10.8043739499474	408	10.3829457755194
9	11.65525985625	59	10.0421936539543	109	9.3040652662	309	10.42159709716	359	10.6602465370447	409	10.3928107251478
10	12.83397913697	60	9.9411719785974	110	7.5800427342	310	10.39564420945	360	10.608496803139	410	10.3963818096224
11	9.890893132787	61	9.51111232566486	111	11.022691241	311	10.49022720348	361	10.5516556256627	411	10.4174959706278
12	11.51215046631	62	9.97076802414598	112	9.3608951625	312	10.47527399315	362	10.4521296301749	412	10.4360562916239
13	10.45963544999	63	10.0406036781028	113	9.5076314321	313	10.37491821426	363	10.4457546302235	413	10.5118779494339
14	8.439287862592	64	11.2103407725836	114	9.7475808624	314	10.36065939308	364	10.4088870483066	414	10.416446526978
15	7.618107159759	65	9.11470827715486	115	9.9357052894	315	10.34185543796	365	10.3047608375247	415	10.3832571259002
16	10.18968002204	66	10.3261788275525	116	10.092922982	316	10.32700788821	366	10.3361520236212	416	11.5732670022091
17	9.660587755657	67	8.56412668301874	117	10.101572899	317	10.18972593316	367	10.3175199042369	417	10.2010326412457
18	9.130679948207	68	10.4252655465527	118	10.098734437	318	10.13805423558	368	10.3147867014108	418	11.4869420413217
19	11.2664304688	69	8.9301681512239	119	10.139759885	319	10.16418460941	369	9.9759453791415	419	9.8585871724023
20	10.28662689597	70	11.1835679738911	120	10.146188575	320	10.19883623384	370	10.0178347504828	420	9.23406506570871
21	8.86576897689	71	10.599982729678	121	10.180866469	321	10.41077199296	371	10.231617305078	421	18.42517043237353
22	10.06522857997	72	10.9284382674003	122	11.016881718	322	10.49921824516	372	10.2872119416505	422	11.5929171908799
23	12.06493459469	73	10.0344117552964	123	10.24223158	323	10.50211681427	373	9.99983103162481	423	11.6196081320245
24	10.9450115242	74	10.2479067321715	124	10.30069088	324	10.44812814411	374	9.78805870465853	424	9.55291054585863
25	10.43069790154	75	9.87674910092658	125	10.179824767	325	10.41308940474	375	10.0586120121154	425	12.2115704574787
26	11.67367809518	76	11.4344527988921	126	10.682191662	326	10.46890635583	376	9.99801062899113	426	11.6431661310986
27	9.302244568737	77	10.8761230995043	127	10.155414941	327	10.48321870794	377	10.0050708481565	427	10.2010326412457
28	7.58895047489	78	9.67499972952709	128	10.559172737	328	10.48818953764	378	10.1053637781831	428	10.4932070207152
29	9.359100488496	79	10.1862418004224	129	10.144071825	329	10.461098056	379	10.2287433542157	429	8.63759244104797
30	9.332669119286	80	9.66916892652912	130	10.129719834	330	10.40732335589	380	10.7542253793957	430	10.3287206913781
31	8.94075473439	81	11.1469926134552	131	10.17516709	331	10.38127947606	381	10.7850494053872	431	9.55237892747025
32	8.757132965309	82	9.97005651743126	132	10.056073167	332	10.40031049272	382	10.6946829292391	432	9.95634738909603
33	10.63864919655	83	11.0149529696705	133	10.178557659	333	10.49030050134	383	10.6703905475006	433	11.5491690467601
34	11.31406445024	84	12.0122954032043	134	10.046994122	334	10.50364283503	384	10.6700591437943	434	10.8256289723335
35	11.90148312721	85	11.4460355112429	135	10.172888585	335	10.43467212676	385	10.5878987996148	435	11.4532202194921
36	9.235407711785	86	8.7573920781899	136	9.643513088	336	10.43932238423	386	10.6636773759523	436	8.93597321896023
37	8.937367567553	87	10.2461725787696	137	10.167445136	337	10.56023831325	387	10.7528394735184	437	10.7947955515884
38	9.641707792726	88	11.7567337355759	138	10.411389745	338	10.6077378646	388	10.8341227530438	438	9.8438488049553
39	10.58496531919	89	9.90578840022547	139	10.180964961	339	10.65349041765	389	10.7864708001881	439	10.0187489874635
40	12.11169393854	90	10.5138418467784	140	9.8586765923	340	10.56779061853	390	10.7804287840956	440	24.6907191346462
41	9.815547860141	91	10.8190686349073	141	10.211751878	341	10.48658143569	391	10.7104942585319	441	10.213758981411
42	11.53111295021	92	8.54936350762606	142	10.820305258	342	10.42074794849	392	10.610068588622	442	9.70563127167602
43	9.486341774111	93	10.8621840676742	143	10.199114744	343	10.51760487593	393	10.4876100077811	443	10.5914417309328
44	8.54087957088	94	9.4144275268702	144	10.390739964	344	10.58516253647	394	10.335590621198	444	11.2224215324731
45	11.40119951615	95	11.8202396073516	145	10.206909479	345	10.64890781818	395	10.1854214146608	445	8.87186727878008
46	11.24095370562	96	10.6080731694589	146	10.391466862	346	10.63640593949	396	11.739647882072	446	9.00249871403531
47	9.671440309397	97	9.61652302969103	147	11.013293731	347	10.5745855301	397	10.1003634607405		
48	9.588246800089	98	11.028431477234	148	10.430935188	348	10.46006923497	398	10.0730720236304		
49	9.660519703944	99	9.34580242838736	149	9.9672129651	349	10.43883126232	399	10.0545744832883		
50	10.38052873148	100	11.3482573460622	150	10.491196321	350	10.47953387159	400	10.1377099945503		

Table 3.2: : Table showing the PSNR count of the salt and pepper noise added dataset

On the later part of this chapter we will take a closer look on this data’s visualization and take a look on the relative analysis on the final features rolling to the model training.

3.4 Feature Analysis

According to the previous discussion the result we generated by using laplacian variance and PNSR of salt pepper noise or any other noise present like gaussian noise. Now we will visualize the data in chart scaling so that it is easier to have the idea about the whole dataset quality and further analysis to come to a conclusion on the features we are using in relevance of the dataset

Firstly, let us take a look of the chart data of laplacian variance of the whole dataset

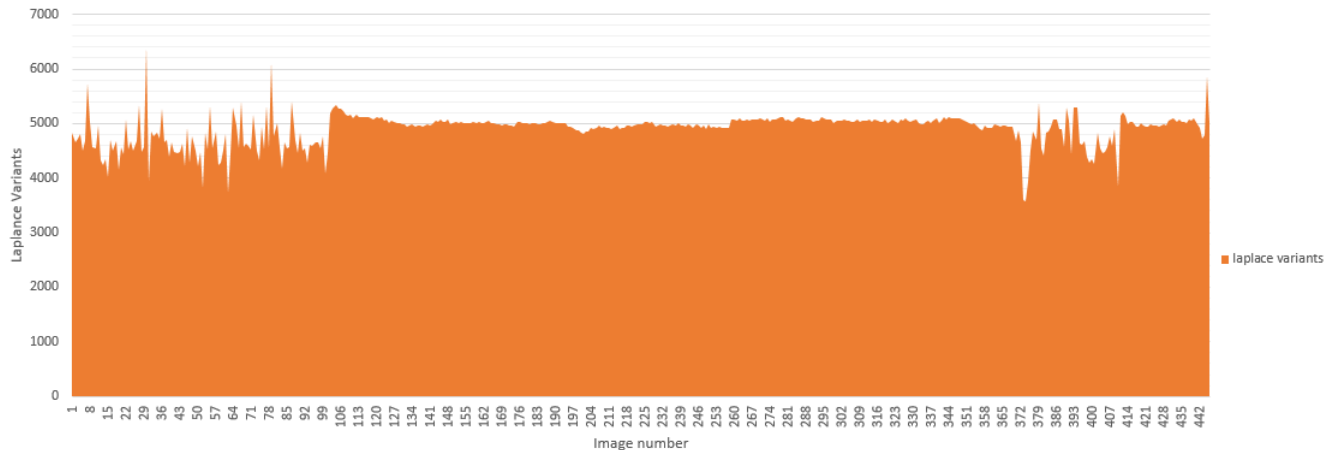


Figure 3.6: : Graph representation of Laplacian variance of the dataset

Here we can see the variance count of the laplacian over a gradual count of images. Keeping the note here that we can see a high peak of the variance that is confirming very blurry image. So on the later part of the work we will be training our Q agent and generate model according to reward biases and again will have a look on the output data. And look for a decrease of the variance to approve our ground truth of decreasing blur in image. But for now we can confirm the feature relevancy according to the laplacian variance of the dataset.

Secondly, Let us have a look on the PSNR analysis of added noise based on salt pepper[First priority] or any other noise like gaussian noise[if previously available]

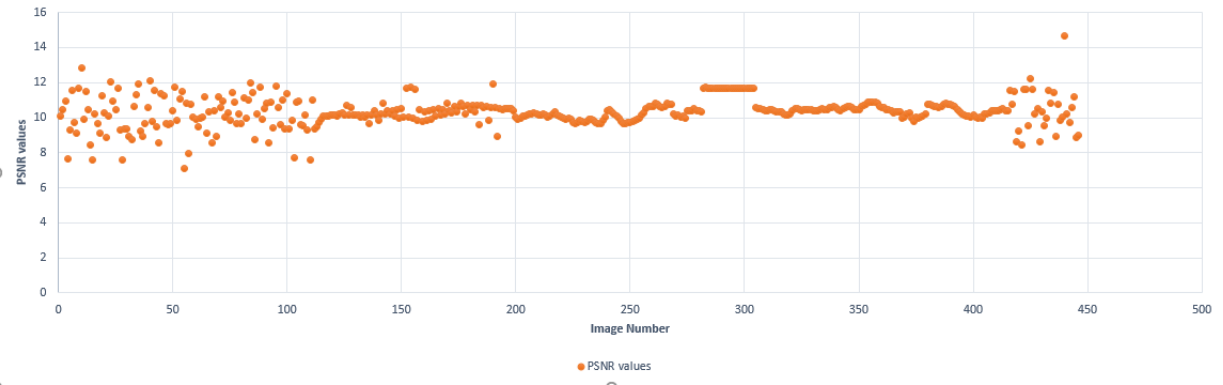


Figure 3.7: : Graph representation of PSNR count of the dataset

Let us have a clear idea what we are seeing here. In the whole chart we can see the PSNR values mapped against the image sequence of the dataset. The lowest value is approximately near 5 and highest value is approximately near 15, [Values have been shown in Figure 3.7. Before the explanation let's know a bit more about how this ratio is working and how we should be expecting the future generation of results to be.

As, PSNR represents a ratio that means $\Delta = x/y$ where x is a standard which represents a non-blurry image and y represents blurry image from dataset. So the resulting Δ or the PSNR value will be less in range of the axis it has been plotted. Or in other words more the noise is less the Δ value.

So the visualized data is confirming the final dataset and the feature with presence of enough noise that needs to be filled by prediction though our Q learning agent. Let's keep a note of this chart and move onto our gradual development of the trained model and the test result and see if this Δ expands it's range and confirms the ground truth of denoising the image dataset.

To conclude after the feature analysis let us use a diagram to overview the process we gradually developed our feature in our dataset. The features has been artificially added so that the ground truth can be obtained after this distorted images are run into the trained model.

Firstly the laplacian variance was compiled and any image not reaching the threshold was excluded. And added noise based on salt and pepper / gaussian noise so that the images can be developed by the intelligent agent.

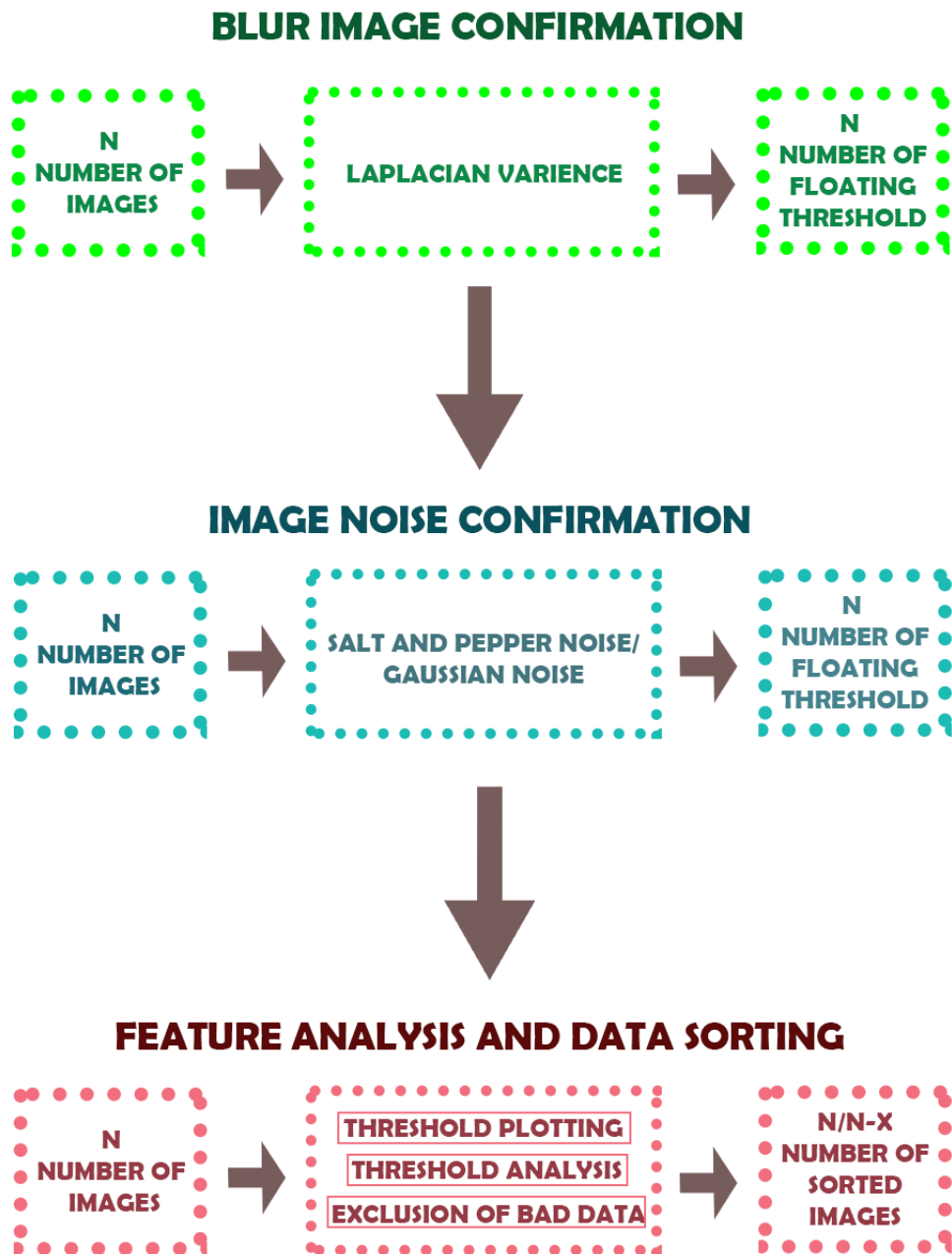


Figure 3.8: : Process overview of adding features

Thus a proper dataset containing suitable features for further training of the model has been generated by following the procedure table of Figure 3.8

Chapter 4

Model Selection and Results Analysis

4.1 Q-Mapping

Q – learning is a reinforcement learning model calculation to learn the nature of activities mentioning to an agent what move to make under what conditions. It doesn't need a specific environment to work on and it can deal with issues with stochastic changes and rewards, without requiring transformations

In the Q-learning algorithm (4) the activity esteem capacity of the ideal approach π^* is iteratively approximated by refreshing the assessed Q-values:

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha \left(r + \gamma \max_a' Q(s', a') \right) \quad (4.1)$$

Here it utilizes recently experienced advances (s, a, s', r) and a learning rate which is signified by α . In ϵ -greedy search, this yield worth can be utilizing to make greedy exploration which is $a = \arg \max_a Q(s, a)$ or irregular activities consistently a $U(A)$ with probability denoted by ϵ . At last, the objective $r + \gamma \max_a' Q(s', a')$ doesn't rely upon the procedure used to deliver the data, permits Q-learning to figure out how to learn off – approach, beneficially re-using past advances set aside in a replay support or made by another segment.[23]

Q – mapping is mainly an advanced version of Universal value function model which takes observation and the goal as an input and creates q – values as an output for every action. But by using q-map it works slightly differently which is it takes an observant as input but generates both goals for every outcome and action as an output. For the observant height and width of each frame may be different from each other according to the requirement of the observation's resolution. All this output works as a Q – value which is mainly a goal reaching reward function where the objective arriving at completing an episode will give reward as 1.[23]

But if it fails to reach the goal i.e., if it doesn't complete the episode then it will get 0. There is a γ which is used as a discount value which creates dramatically decaying esteems $\gamma^{(k-1)}$ which express the quantity of steps k to the objectives. Starting, a

forward pass which is in the model is compiled with the accompanying perception in data and afterward the made framework are augmented on the movement estimation to deliver a solitary edge addressing an extent of the base foreseen quantity of steps towards all of the goals. Now the edge is cut in the range $(0, 1)$ to lessen the under and over appraisal. At last, the edge is restricted by γ and the territory came to at the accompanying stage is set to 1.

This system sufficiently utilizes one common advancement to basically make a lot of autonomous one episode, one for every objective, with an end point on success and halfway scene bootstrapping [22]. To create Q-frames to arrive at an objective, one just needs to take the vector of Q-values at the objective area and select the movement of most outrageous worth.

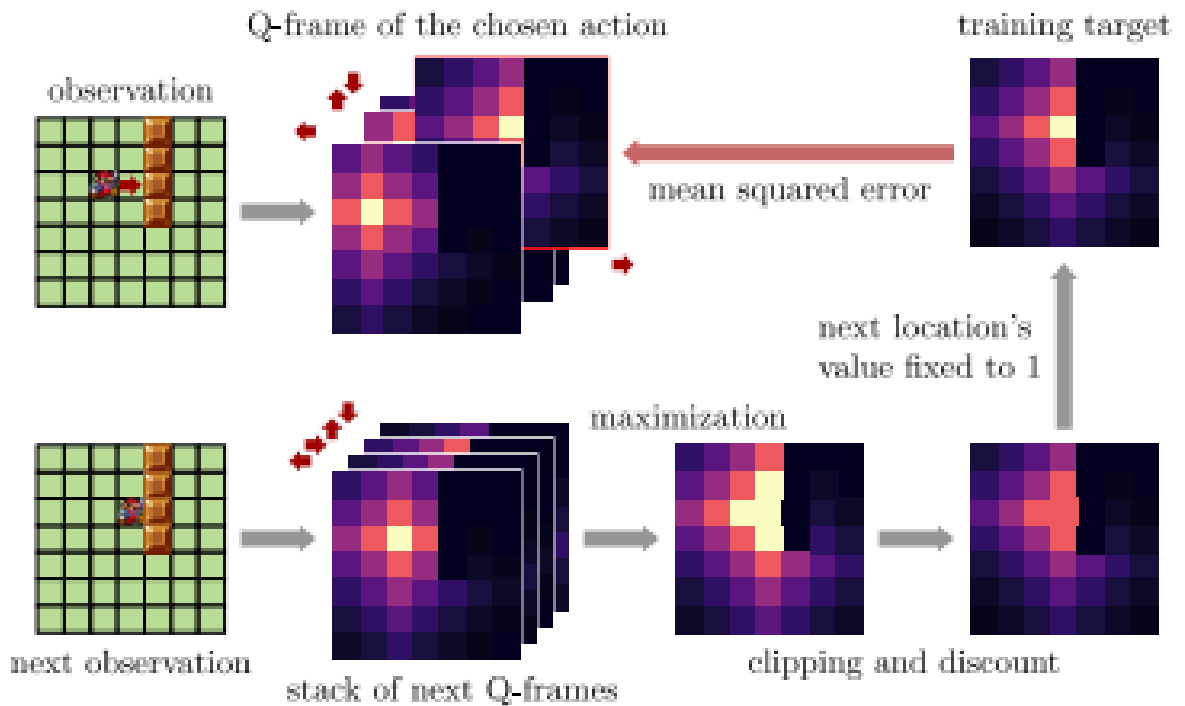


Figure 4.1: Training process for the proposed model, updating the prediction towards all goals at once.[23]

4.2 Asynchronous Advantage Actor Critic(A3C)

A3C is a kind of algorithm where there is a global system where numerous specific agents which each have their own arrangement of network parameters. Every one of these agents cooperates with their own duplicate of the environment simultaneously as different agents are communicating with their surroundings. The main reason this process works in a way that is better than having a solitary agent, is that the experience of every agent is free of the experience of the others. In this manner the general experience accessible for preparing turns out to be more assorted. [25]

So in this algorithm it keeps a policy $\pi(a_t | s_t; \theta)$ also, a measure of a value esteem function $V(s_t; \theta_v)$. This algorithm is fairly irregular on the grounds as it works in the forward view by unequivocally computing n step returns, rather than the more normal in reverse. [2] To register a solitary update, the calculation initially chooses action utilizing its exploration strategy for up to t_{max} steps or until a terminal state is reached. This cycle brings about the specialist accepting up to t_{max} rewards from the environment since its last update. The calculation at that point registers angles for n-step Q-learning refreshes for every one of the state-activity sets experienced since the last update. Every n-step update uses the longest conceivable n-step return bringing about a one-step update for the last express, a two-step update for the second keep going state, etc for a sum of up to t_{max} refreshes. The gathered updates are applied in a solitary inclination step. [20] The update performed A3C can be seen as

$$\nabla_{\theta} \log \pi(a_t | s_t; \theta') A(s_t, a_t; \theta, \theta_v) \quad (4.2)$$

Here $A(s_t, a_t; \theta, \theta_v)$ is a gauge of the preferred position function given by

$$\sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V(s_{t+k}; \theta_v) - V(s_t; \theta_v) \quad (4.3)$$

Here k will change for every step.

Similarly, as with the worth put together strategies, we depend with respect to resemble actor learners and amassed refreshes for improving training stability. As the boundaries θ of the arrangement furthermore, θ_v of the worth capacity appear as being isolated for over-simplification, we generally share a portion of the boundaries in practice. We commonly utilize a convolutional neural organization that has a specific softmax yield for the strategy $\pi(a_t | s_t; \theta)$ and a direct yield worth capacity $V(s_t; \theta_v)$, with all non-yield layers shared.

We likewise found that adding the entropy of the policy π to the target work improved investigation by debilitating untimely combination to imperfect deterministic strategies [1]. The gradient of the function combining entropy regulation with the requirement of policy parameter takes the form

$$\nabla_{\theta} \log \pi(a_t; \theta') (R_t - V(s_t; \theta_x)) + \beta \nabla'_{\theta} H(\pi(s_t; \theta')) \quad (4.4)$$

4.3 Pixel Reinforcement Learning

PixelRL is also one of the effective learning methods. In this method every pixel has its own agent and that agent changes the pixel esteem by taking an activity. It additionally has a compelling learning technique that fundamentally improves the presentation by thinking about not just the future conditions of the own pixel yet additionally those of the neighbour pixels. It can be applied to a few picture handlings errands that require pixel-wise controls, where profound reinforcement learning has never been applied. Plus, it is conceivable to envision what sort of activity is utilized for every pixel at every cycle, which would assist us with getting why and how such an activity is picked. We additionally accept that our innovation can improve the logic and interpretability of the profound neural organizations. Furthermore, in

light of the fact that the tasks executed at every pixel are envisioned, we can change or adjust the activities in the event that is vital. [24] In this section we portray the proposed PixelRL issue setting. Let us take li the i^{th} pixel which is utilize as an input image A that contain a set of N number of pixels so the series extend from 1 to N. Each pixel is backed by their own agents and the policy that are contained for them is denoted as

$\pi_i \left(a_i^{(t)} \mid s_i^{(t)} \right)$ and in this expression $a_i^{(t)} (\in A)$ and $s_i^{(t)}$ are the activity and the constraints of the i^{th} agent at the given time t. A is the pre-characterized activity set, and $s_i^{(0)} = A_i$. The agents get the following phase $s^{(t+1)} = \left(s_1^{(t+1)}, \dots, s_N^{(t+1)} \right)$ and rewards are stated as $r^{(t)} = \left(r_1^{(t)}, \dots, r_N^{(t)} \right)$ from the created situation by making the moves $a^{(t)} = \left(a_1^{(t)}, \dots, a_N^{(t)} \right)$ [25] The goal of the pixelRL issue is to become familiar with the ideal strategies $\pi = (\pi_1, \dots, \pi_N)$

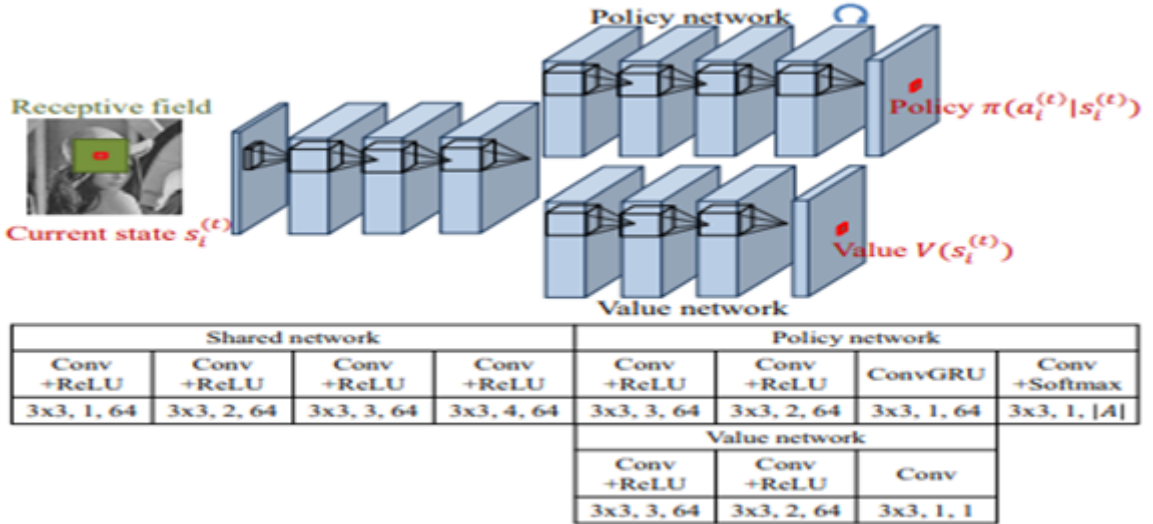


Figure 4.2: Network architecture of the fully convolutional A3C. The numbers in the table denote the filter size, dilation factor, and output channels respectively. [24]

For which it maxed the mean of all the rewards that can be obtained by all pixels.

$$\pi^* = \operatorname{argmax}_{\pi} E_{\pi} \left(\sum_{t=0}^{\infty} \gamma^t \bar{r}^{(t)} \right) \quad (4.5)$$

$$\bar{r}^{(t)} = \frac{1}{N} \sum_{i=1}^N r_i^{(t)}$$

In this equation $r^{(t)}$ represents the rewards of the pixel and $r_i^{(t)}$ represents the pixel. Another arrangement is to isolate this issue into N autonomous subproblems and train N networks, where we train the i^{th} agent to boost the normal absolute compensation at the i^{th} pixel:

$$\pi_i^* = \operatorname{argmax}_{\pi_i} E_{\pi_i} \left(\sum_{t=0}^{\infty} \gamma^t r_i^{(t)} \right) \quad (4.6)$$

A simple method to settle this issue is to set up an organization that yields Q esteems for all possible game plan of exercises $a(t)$. But preparing N number of networks is in like manner computationally impossible at the time when the quantity of pixels is colossal. In addition, it treats simply the fixed size of pictures. For this we utilize FCN where all the N number of agents can equally utilize the boundaries, and we can parallelize the count of N number of agents on a GPU, which delivers the testing capability.[24]

Pixel RL is essentially unique in relation to multi agent RL issues for some terms. One of the reasons is the number of agents. The number of agents required are an extremely huge amount so multi agent technique cannot be applied directly in pixel IRL.

4.4 Edge Detection for blur reduction

Another method that is used in this research is edge detection/finder method. An edge finder is a straightforward calculation to separate edges from a picture. Applying an edge identifier to a picture creates a bunch of associated curves which follow the limits of an object. Most edge finder methods utilize either first order derivatives or second order derivatives. [5] By first derivatives to find the local edge direction it gives the direction of the local maximum of a point. By the second order derivatives are used to locate local maxima of a gradient. These maxima can be found from zero crossing of Laplacian or nonlinear differential expressions. [8] This method mainly works by detecting the change of intensity between two close points. The higher the intensity difference the more accurately we can detect the edge between those points. There are many types of edge detection methods where we use the Sobel edge detection type.[8] By this we attempt to accomplish greater adaptability and versatility with our exploration.

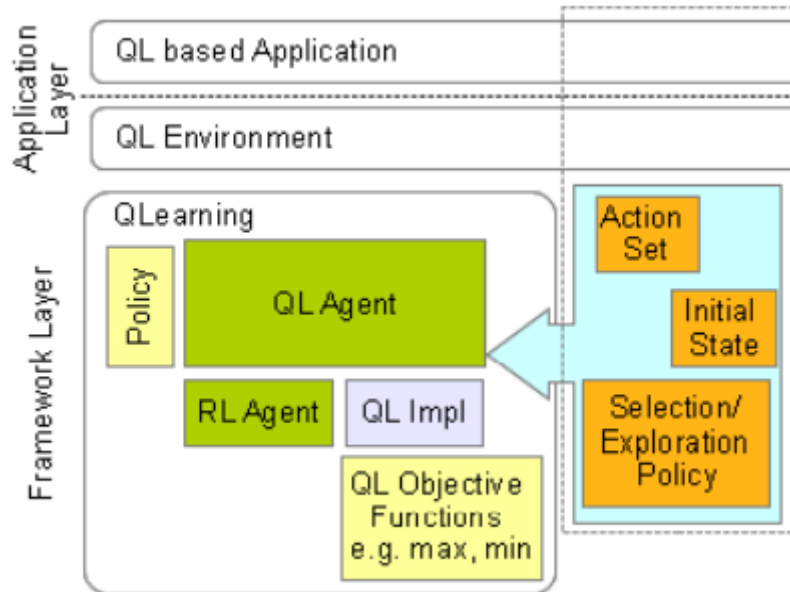


Figure 4.3: Q-Learning framework architecture. [8]

In figure 4.3 we try to show the framework of our Q – learning algorithm where there is a Q – learning agent (QL), a reinforcement learning agent (RL), Q- learning implementation (QL), there is a policy containing the data structure of the system. It contains the learning rate and discount factor of the algorithm. For this algorithm there are many types of activity to detect edges for every step of the calculation so in the action set it holds all the action it needs to perform. Also, there is an initial state which we need to implement and cannot be changed later. Selection and exploration policy control the policy inside the framework which policy will be active in which case/ environment.[8]

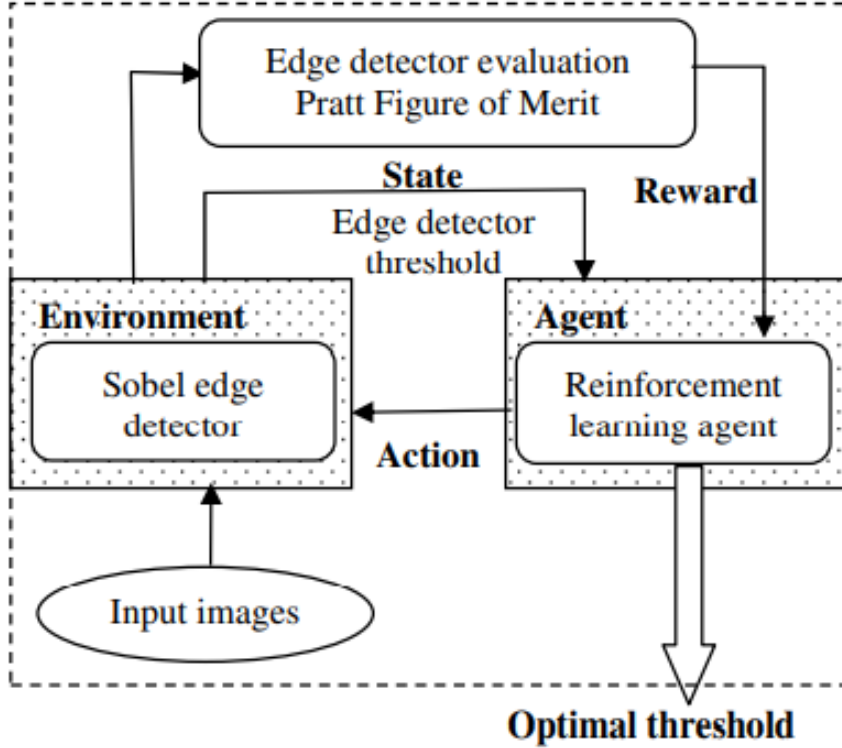


Figure 4.4: The proposed automatic parameterization for the Sobel edge detector. [8]

In figure 4.4 we have shown the working procedure of the Sobel edge detector method that we used in this research. This method mainly figures out how to locate the optimal edge discovery edge for numerous pictures. States are spoken to as qualities for the looked through boundary the only two potential activities are addition and decrement of state’s worth and rewards are registered utilizing Pratt Figure of merits [11]. The Sobel edge detector takes edges from the pictures utilizing an edge esteem given by the reinforcement learning agents. The separated edges are evaluated by the assessment module and the outcome is used as a compensation for the reinforcement learning agents, which takes the related action and gives another edge detector threshold. The learning measure proceeds until the ideal limit is found. Pratt Figure of Merit is a notable measure used to analyse edge detectors output. It tries to adjust three sorts of mistakes that can create wrong edge mappings: missing legitimate edge points, disengaged edge points and misclassification of commotion variances as edge points. Pratt Figure of Merit utilizes a known edge for correlation [11]. We consider a known edge the edge came about because of the output of a Canny edge locator dependent on a manual picked edge. For various edge recognition edges, we outwardly examine the subsequent edges for the Canny detector and we picked the limit that accomplishes the ideal conceivable edge. The figure of merit [11] is defined as –

$$K = \frac{1}{I_N} \sum_i^{I_A} \frac{1}{1 + \alpha d_i^2} \quad (4.7)$$

In this equation 4.7 I_A represents the summation of all test edge pixels and I_i presents the number of known edge pixels in the picture. α is a scaling consistent, while d_i is the distance from a genuine edge highlighting the closest known edge point. The scaling factor is utilized to penalize edge points that are bunched however away from the known edge pixels. After exploring different avenues regarding different qualities, we chose to utilize $\alpha = 0.9$. This value is prepared by Pratt in [8], a value that drastically penalizes pixels far away from the known edge. Q-Learning algorithm updates its Q-values using:

$$Q(s_t, a_i) = Q(s_t, a_i)(1 - \alpha) + \alpha [r_{i+1} + \gamma_a^{\max} Q(s_{t+1}, a)] \quad (4.8)$$

Here r_i is the rate of reward at time t . Also γ is the discount factor range between 0 and 1. We used greedy policy in this paper. The learning rate is utilized to set up the impact of the new gained data against past encounters. A little factor will cause the agent to learn next to none and depend exclusively on past experience gained or given as info premises, while a major factor would cause the agent to consider just the latest data. The discount factor builds up the effect of potential compensations. An invalid factor will cause the agent to think about current prizes, while a major factor will cause it to accomplish a high long-term reward.

Finally we have these four methods *Qlearning*, A3C, Pixel RL and Edge detection methods that work parallelly to each other. Like in Q - learning it mainly works as a reward based system which can be received by reaching at the goal of a function. A3C is a method or algorithm which handles the functionality of a multi agent network and see what each of them accomplished in one run.[8] Pixel RL is another interesting method where it handles each and every pixel with a specific agent and tries to manipulate each pixel that can be extracted from an image and this improves the visual presentation of an image. And lastly Edge Detector method which extracts edges or curves of an object from an image. This method is widely used in the area of image processing. By using this method we get good and accurate boundaries of an object from a blurred or mis focussed image we used in our data set. So all through these models we tried to restore an image from some specific corruption like noise, blurred image and mis focus image. Each model has their own advantage which combinedly works to create or restore a broken or corrupt image more efficiently by using reinforcement learning.

4.5 Result and Analysis

4.5.1 Reward Test results

After we are done with training the model, the output trained model is then run with respect to the test dataset which contains noisy and blurred images. Let us first have a look on the extracted data for each 100 episodes upto 1000 episodes which has been plotted into the following graphs :

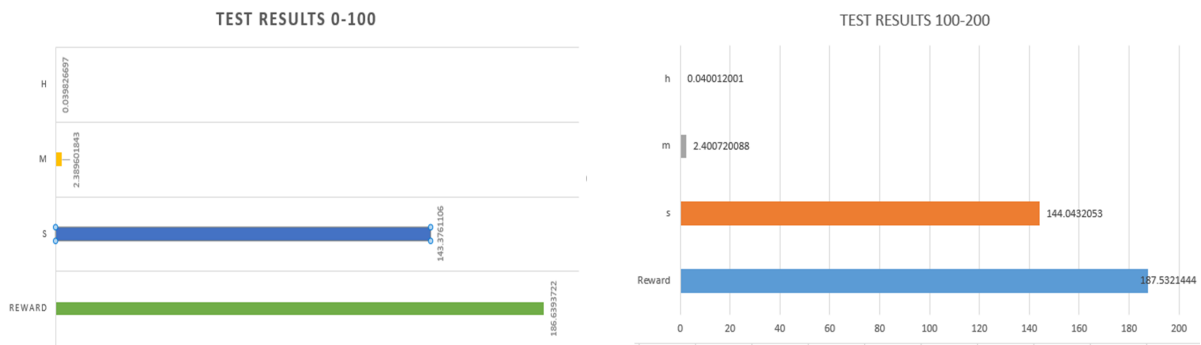


Figure 4.5: Reward Test result of 0-200 episode

Figure 4.5 shows reward Test result of 0-100 and 100-200 which shows initial rewards generated from the training. The initial rewards are not that efficient as they still lack enough exploration.

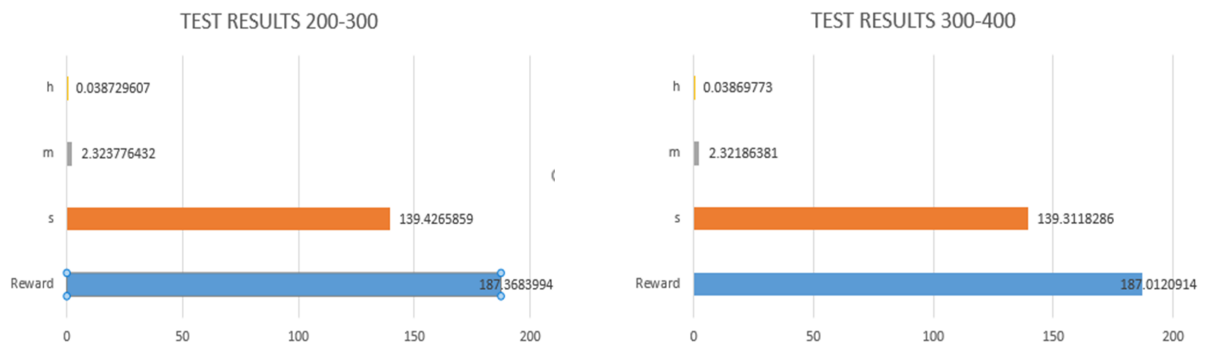


Figure 4.6: Reward Test result of 200-400 episode

Figure 4.6 shows reward Test result of 200-300 and 300-400 which shows increase in rewards. And consistency in increase in exploration

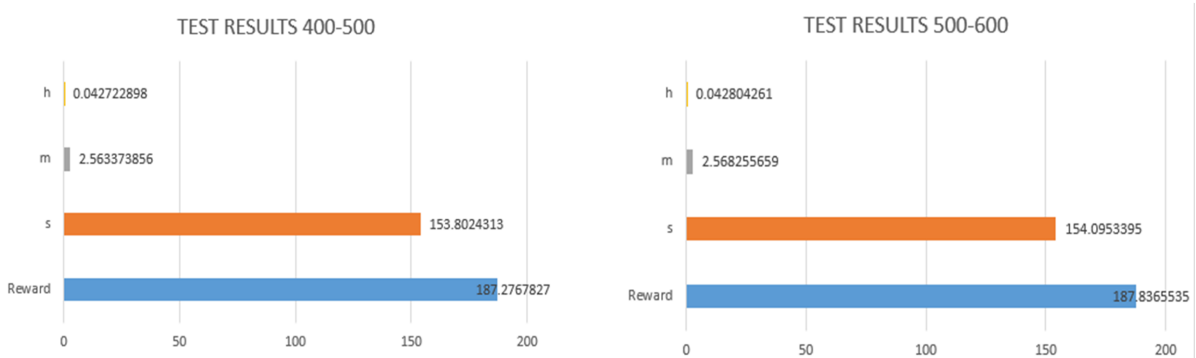


Figure 4.7: Reward Test result of 400-600 episode

Figure 4.7 shows reward Test result of 400-500 and 500-600 which shows increase in rewards and consistency in exploration, but the output result needs more development

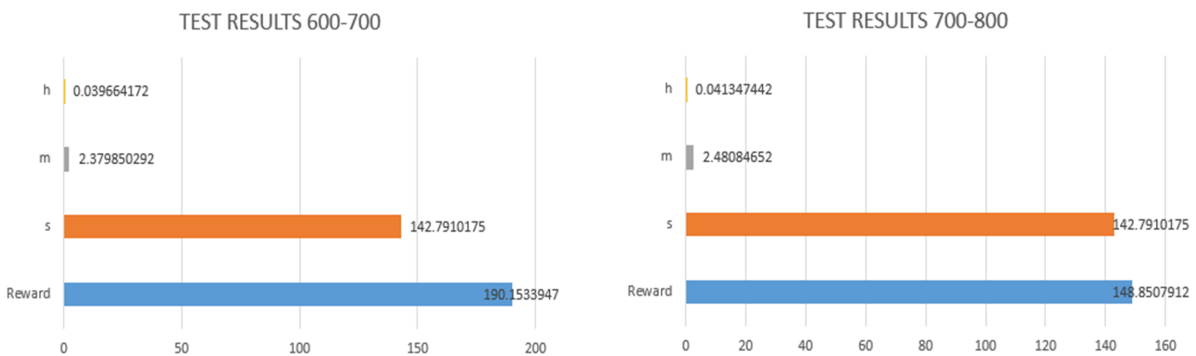


Figure 4.8: Reward Test result of 600-800 episode

Figure 4.8 shows reward test result of 600-700 and 700-800 which shows increase in rewards and consistency in exploration, and the output is getting better into optimal

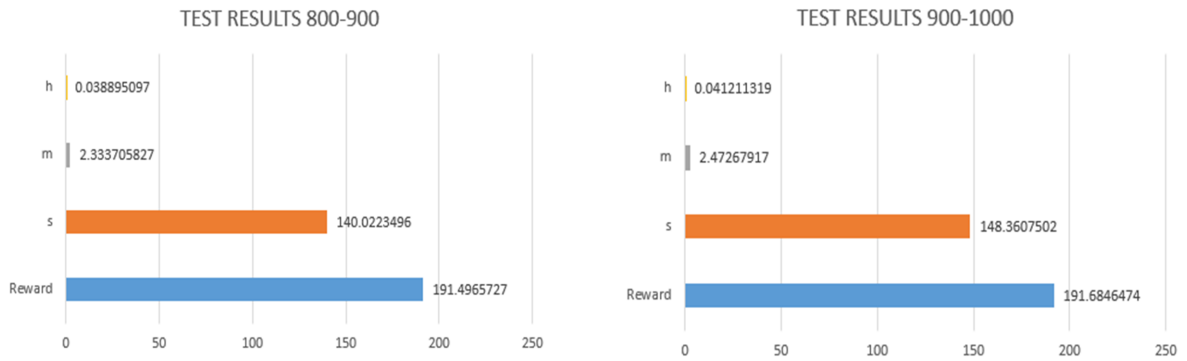


Figure 4.9: Reward Test result of 800-900 and 900-1000 episode

Figure 4.9 shows reward test result of 800-1000 900-1000 which shows dramatic increase in rewards and consistency in exploration, and the output is fairly recognizable as a developed image

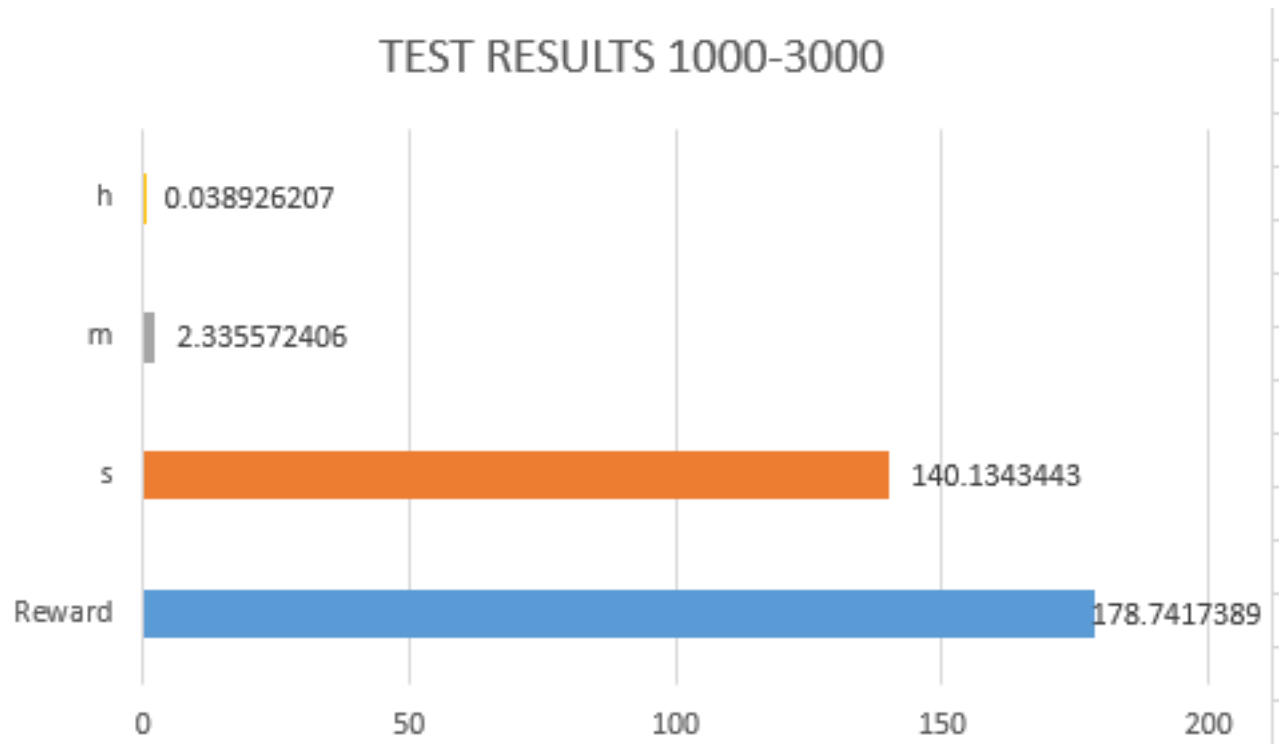


Figure 4.10: Reward Test result of 1000-3000 episode

Figure 4.10 shows reward test result of 1000-3000 which shows a drop in reward but the output generated model has become more efficient as the Q map has expanded and the pixel wise AI agent has explored more of the train dataset iteratively. The increase in efficiency can be observed in the later part of this work in the section 4.7

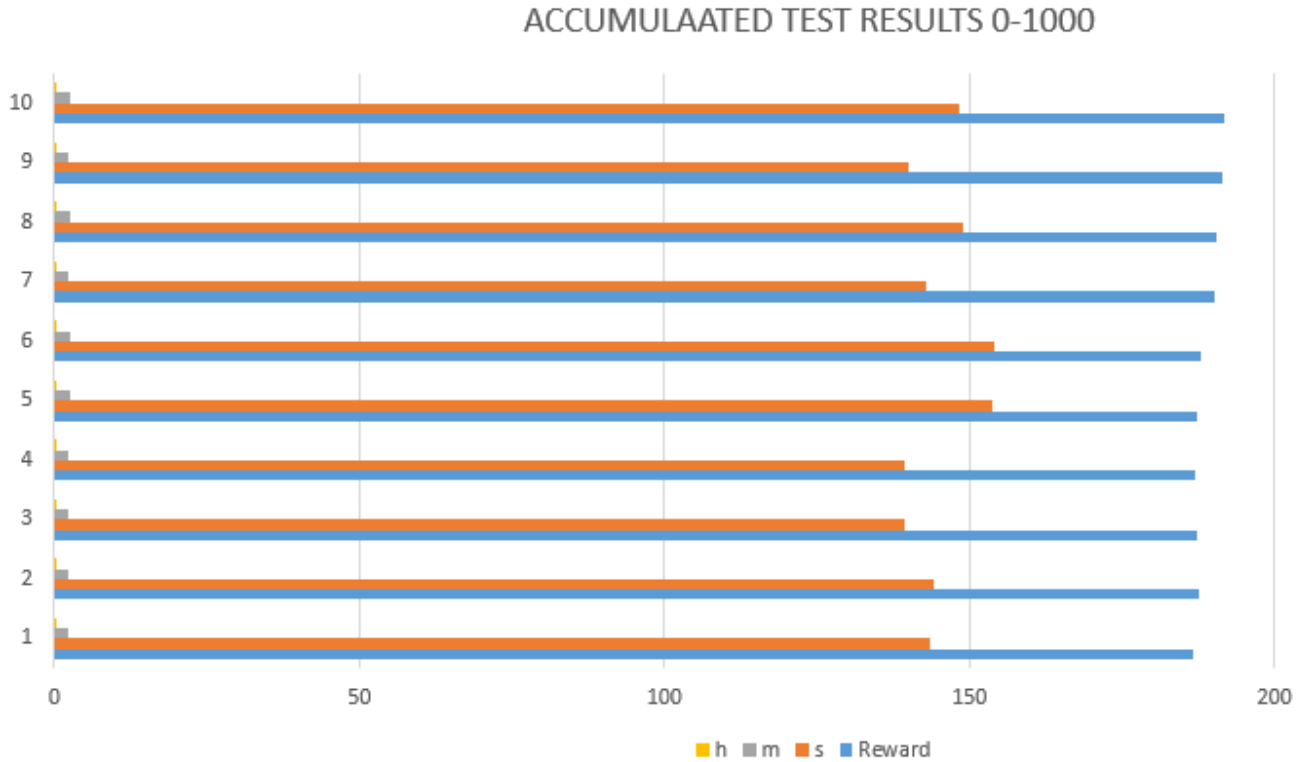


Figure 4.11: Superposition of Reward Test results of 0-1000 episode

Here , as we have shown the breakdown of the result of first 1000 episodes into gradual development based on the proposed model , we first plotted the first 1000 episode into batch(100 in each) so this generated us 10 subsequent results and he in this Figure 4.12 we have made a superposition of their performance based on reward exploration which confirms development ratio of the image . That in other word means the reward bias is increasing fairly and the efficiency of the model is getting better with each batch of training episodes . We will have a look on their picture restoration performance on the later part of the work.

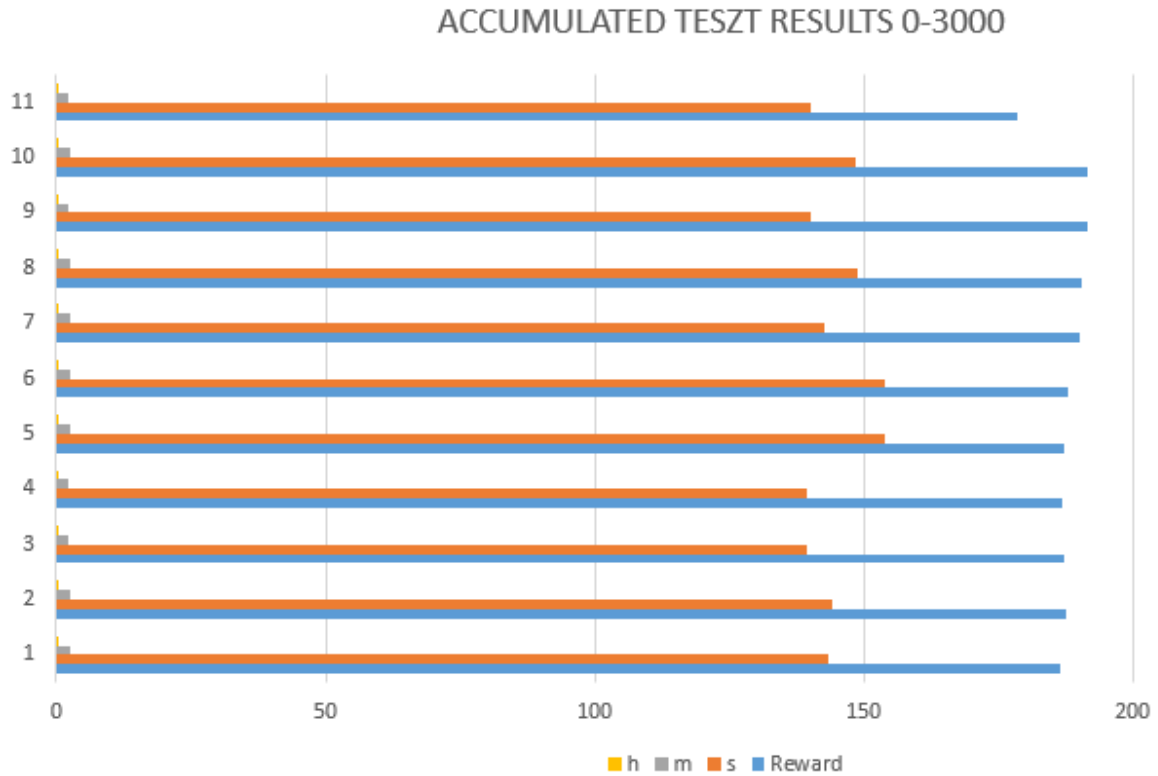


Figure 4.12: Superposition of Reward Test results of 0-3000 episode

As the batch training goes on the model develops more efficient reward bias and a Q map. The end result is an optimised model which helps in more efficient recognition of the high laplacian variance affected areas and noise. Thus also resulting in a more polished picture as the AI agent for each pixel now knows that previous explorations of dataset images. We plotted the performance again after 3000 episodes to observe any dramatic changes as the batch training is stretched further, here in Figurexx the 11th row represents upto 3000 batch episode results, here the reward bias drops but the agent itself now can get more satisfactory result because this has accumulated more accurate data thus the bias drops here and on training further it will gain the reward bias again. But performance has been recorded high at this stage. Confirming the development of the reward bias and development of the input subject images to restored output images we will now move on to our next stage of image quality confirmation using PSNR test on the batches to ensure the quality of the grout truth of our work that is the restored images will retain object clarity and identity in it.

4.6 PSNR Test results

After carrying out the reward test analysis , just like we ran our PSNR test result in the data pre-processing and feature analysis chapter , we will be using the PSNR test again in the output images of the trained models based on batch episodic training. And observe the changes in PSNR data visualization to come to a conclusion if the model produced a satisfactory result. We have discussed before that the PSNR is a ratio(Peak Signal to Noise Ratio) that means the represents the ratio of noise between a standard noise free image and a noisy image. So we will be comparing the Input noisy image with output noise reduced image and visualize it in a graph to observe performance.We have to remember that, less the noise more will be the result of ratio. That means we had a highest threshold of 15 and we need to observe a high scale more than that in the test result.

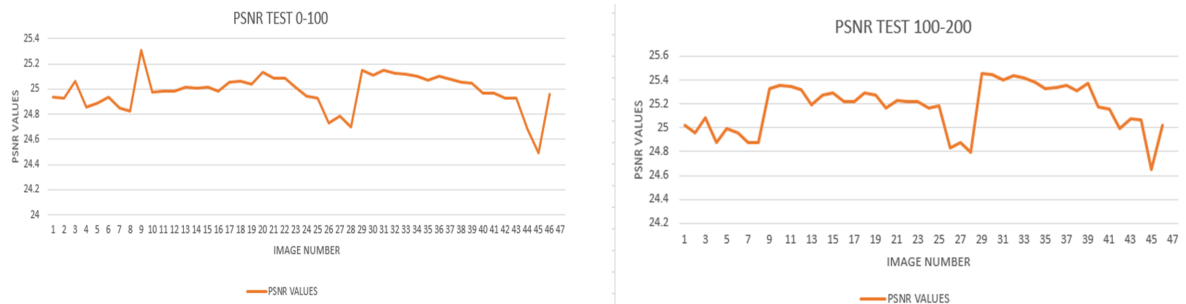


Figure 4.13: PSNR Test results of 0-200 episodes trained model output image

Figure 4.13 shows PSNR Test result of 0-100 100-200 which shows the output images generated a higher threshold on the PSNR count in comparison to the initial dataset PSNR count confirming the reduction in noise.

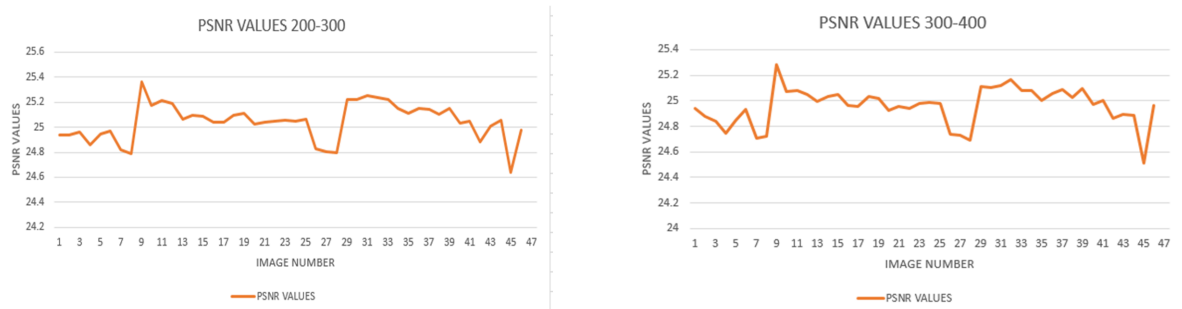


Figure 4.14: PSNR Test results of 200-400 episodes trained model output image

Figure 4.14 shows PSNR Test result of 200-300 300-400 which shows the output images generated a similar threshold on the PSNR count in comparison to the previous PSNR count confirming the continuation of reduction in noise.

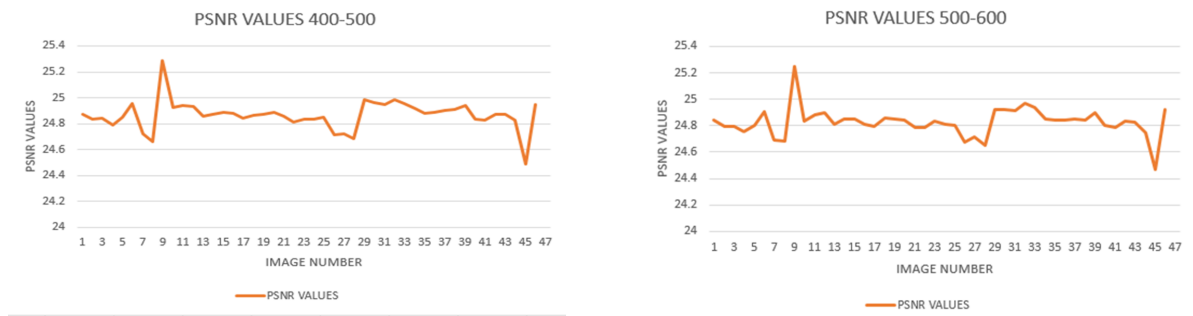


Figure 4.15: PSNR Test results of 400-600 episodes trained model output image

Figure 4.15 shows PSNR Test result of 400-500 500-600 which shows the output images generated a similar threshold on the PSNR count but the over all limit has gone up, that means the over all dataset has been developing more than before and the Pixel wise Ai agents is more efficient now.

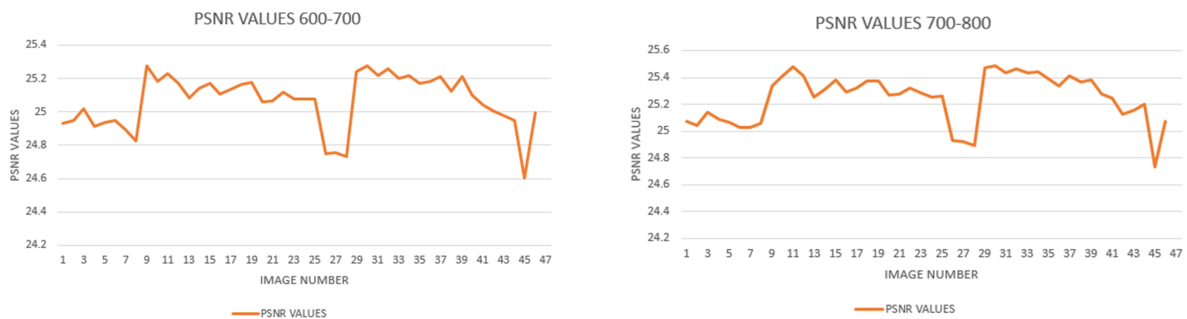


Figure 4.16: PSNR Test results of 600-800 episodes trained model output image

Figure 4.16 shows PSNR Test result of 600-700 700-800 which shows the output images generated a similar threshold on the PSNR count but the over all result of test dataset has been increasing satisfactorily and the generated images is recognisable fairly that we will have a look in the later part of the chapter.

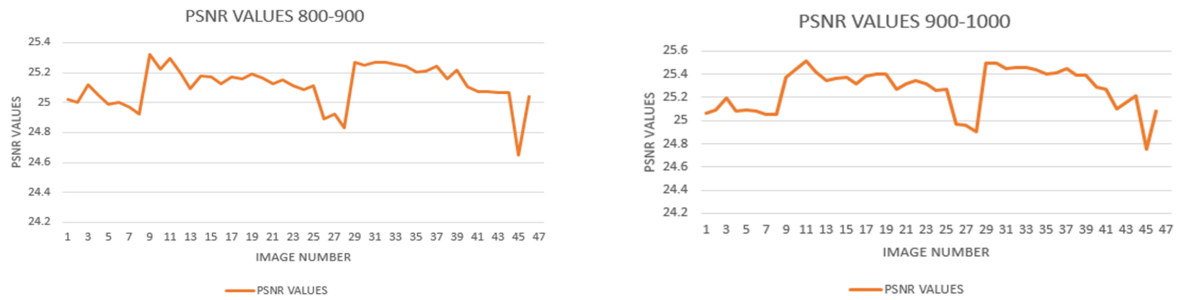


Figure 4.17: PSNR Test results of 800-1000 episodes trained model output image

Figure 4.17 shows PSNR Test result of 800-900 900-1000 which shows the output images generated a higher threshold on the PSNR count and also over all result of test dataset has been increasing satisfactorily and the generated images is recognisable fairly that we will have a look in the later part of the chapter.

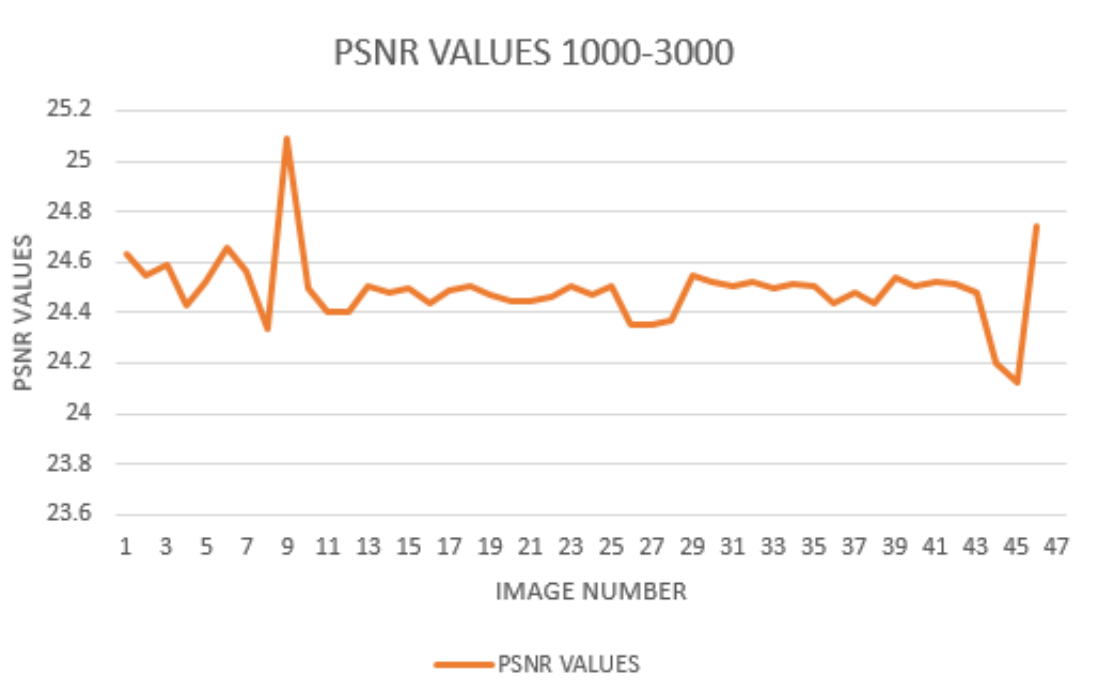


Figure 4.18: PSNR Test results of 1000-3000 episodes trained model output image

Figure 4.18 shows PSNR Test result of 1000-30000 batch which shows the output images generated a moderate threshold on the PSNR count which is almost similar to the inherited graphs of the previous performances. But , there is a dramatic result in the output images. Since the AI agents are more aware of the previous explorations , even in this similar PSNR threshold the result image generated was

mostly restored and smoothed out bringing out the identity that satisfies our ground truth of our work. We will have a look at the generated image on the later part of this chapter. And conclude if the performance is satisfactory.

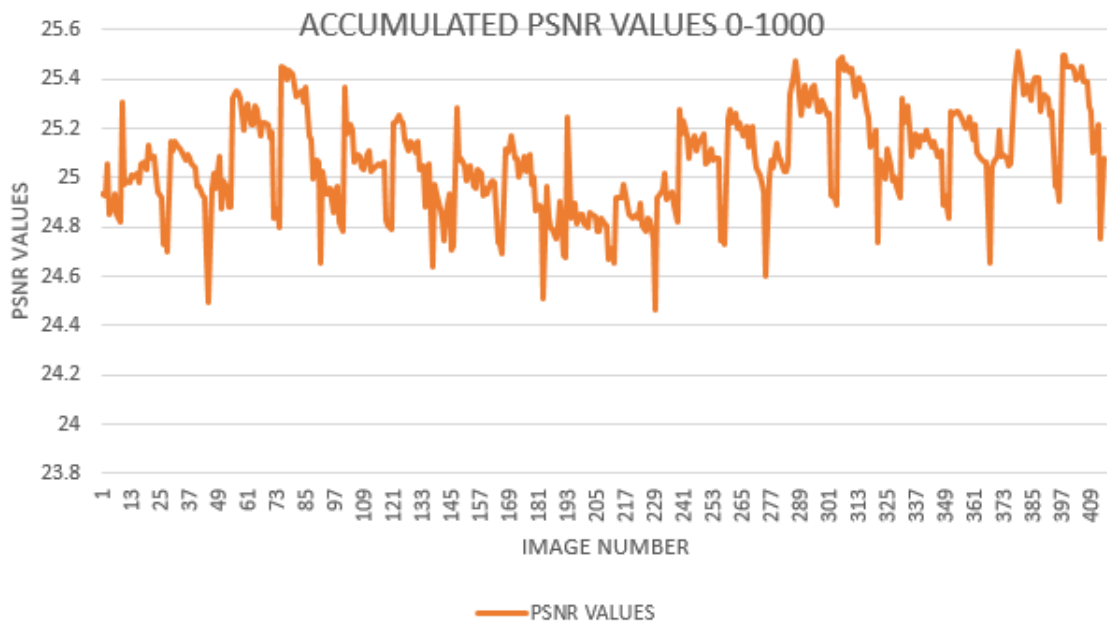


Figure 4.19: ACCUMULATED PSNR VALUE 0-1000

Figure 4.19 shows Superpositioned PSNR Test result of 0-1000 batches which shows the output images generated in a whole was within a higher PSNR threshold which clarifies that the output images were noise reduced and the gradual development of the exploration of the agent was stable. Thus it is also clear that for better output results the agent can be trained further and observed the performance to properly clarify our ground truth.

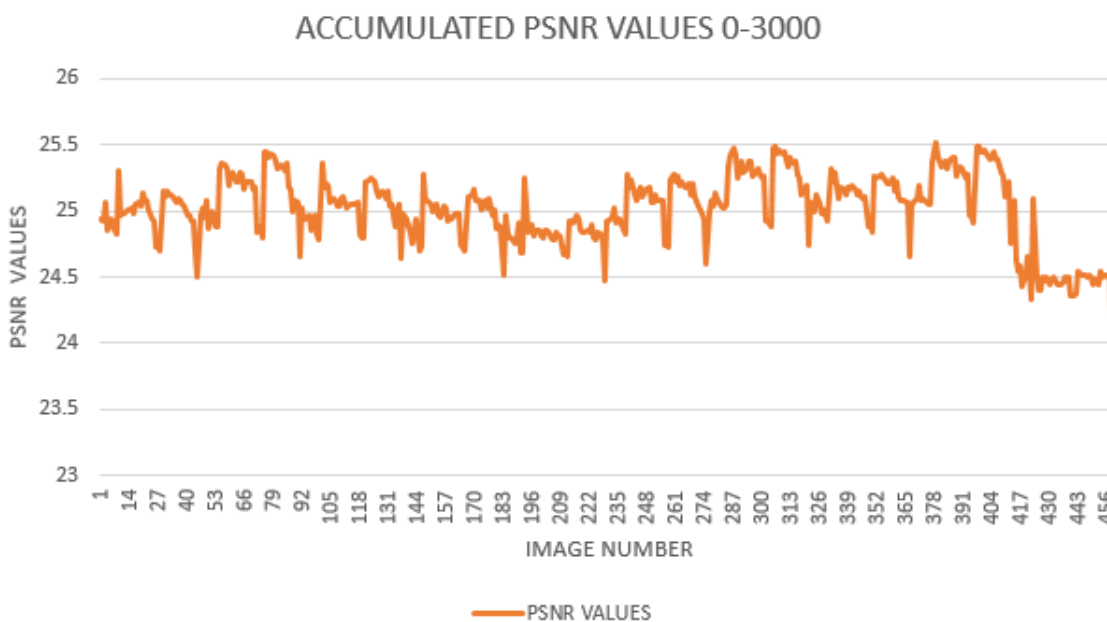


Figure 4.20: ACCUMULATED PSNR VALUE 0-3000

Figure 4.20 shows After superpositioning the PSNR Test result of 0-3000 batches we can observe consistency but the output result was dramatically changed as we have mentioned before the pixel agents have exploited the dataset more than before and the Q maps has expanded dramatically resulting in better model that can generate efficient output restored images. We will be having looks and analysis on the output images on the following part of the chapter.

4.7 Result analysis and Result summary

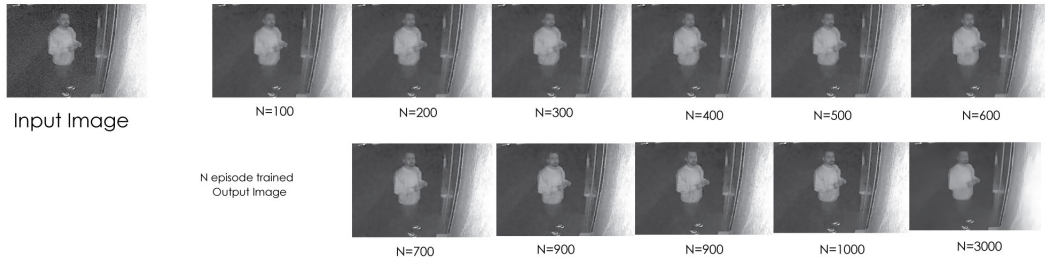


Figure 4.21: Final Result

shows In the figure 4.21 we have represented an image from the test dataset in the left side named as the input image. And on the right side we can see the relative output of the image going through the model trained upto N episodes in batches. The gradual development of the performance is noticeable over each batch of episodes. And specifically there is a dramatic change in the episode 3000 as the model was well trained due to the expansion of the Q map and more exploration of the past dataset. The Pixel wise AI agent has shown remarkable output over time and episode. But keeping in note that this model generated exponential memory complexity for which the result was carried out upto 3000 episodes. But within this 3000 episodes the identity was clearly visible and noise was reduced at a very considerable rate. Also after arranging the results, the blur edges of the image were restored and smoothed bringing the identity more clearly visible.

Finally analyzing the facts based on the output image results and all the data represented in the previous sections of the chapter we can conclude that the ground truth of our work that is enhancing the object clarity by removing the noise and blur edges using the proposed model which is deep reinforcement learning which also utilizes the Q-learning model has considerably enhanced the clarity of images extracted from single channel image or video source.

Chapter 5

Conclusion and Future Work

Image clarity enhancing using classical image processing and machine learning has already been an existing practice. But the limitations of these processes includes time complexity , efficiency and lack of intelligence on building up identity from distorted images. Classical image processing implements basic mathematical models that can not stretch the features out of an image. And Machine learning is based on biased data learning which requires an adequate amount of dataset and classification. But there are many scenarios where generating that adequate dataset is critical and predictions need to be made within limited learning. One of these scenarios is restoring distorted images from an environment specific dataset. In most of these cases Machine learning can not predict without pre trained data and there are limited models to solve specific problems. Moreover the training requires a huge amount of good dataset.

In these cases the study and implementation of AI based approaches is getting more emphasis than ever before. So we approached with the initial idea of implementing AI in image processing , to be specific image restoration. Which can open new possibilities. One such area to solve was restoring single channel night vision images from cctv footage where crucial data need to be retrieved and in many cases the identity in an image may need optimization to extract specific events or identity for various purposes. Starting from criminal identification to accidents and low vision events. Where image clarity and noise can distort to a great deal of extent.

All these facts and studies inspired us to implement AI agent in image pixel processing so that an agent can be trained in a limited data environment and a Q-map can be implemented for that particular agent properties which will result in completing the structure for exploration and exploitation of the dataset.

With all that study and hypothesis we started working with a novel approach, based on Pixel Reinforcement Learning. Initially we approached a Hierarchical image processing model to build the Deep Reinforcement Learning environment. With that being implemented we further implemented the Q-Map model and the architecture of generating the trained output model so that it can be tested in future events.

The whole process faced a lot of optimization because the architecture of reinforcement learning generated an exponential time and space complexity which was

further needed to be optimized by tuning the layered architecture.

But after all these there still remained few backlogs due to space complexity. As we had a limited access to hardware. The theoretical and the practical requirement for the processing exceeded the limit of memory to allocate 80281600 bytes (total 5234877440 bytes). The processing architecture was based on CUDA and Cudnn. And the physical architecture uses Nvidia Tensor core or in alternative only the CUDA core. All the results generated were benchmarked in Nvidia Geforce GTX 1060 with a memory capacity of 6 Gigabyte.

So, we had to train them in batches and compile the results. Moreover the single channel images needed to be pre processed from the cctv images to make sure the agent's do not learn any false data as the environment is already limited to data sources.

Starting from the hypothesis to implementation after all the preparations the model was successful to denoise and develop the image which is noisy and blur. Using the deep reinforcement learning in single channel night vision images. Moreover after the optimization of the system the model ran in a considerable amount of memory complexity.

The major benefit of building up such a model was the AI agents can be trained in a very limited dataset that in our case is almost as low as 500 images. The system is faster in predicting. And the learning procedure is carried out by batch episode of iterating the dataset by the agent. In a whole the model took less time to be trained. It was fast and robust in case of testing / generating output. The concept of using the agents to develop the image was successful in all claimed scenarios in efficient ways.

The results and the approach showed promising outcomes and the future implementation and research are vast in this field. The limitations need to be optimized and we worked on the specific fields of cctv images. But we are very optimistic about implementation of this AI model in dynamic video processing. Video clarity optimization in specialized scenarios. This model showed such an efficiency that we assume this model can be further tweaked and implemented in bigger image processing complexity such as torn/missing image segment restoration , predicting identity and many more endless opportunities by further research.

Bibliography

- [1] R. Williams and J. Peng, “Function optimization using connectionist reinforcement learning algorithms,” *Connection Science*, vol. 3, pp. 241–, Sep. 1991. DOI: 10.1080/09540099108946587.
- [2] J. Johnson, J. Li, and Z. Chen, “Reinforcement learning: An introduction: R.s. Sutton, a.g. Barto, MIT Press, Cambridge, MA 1998, 322 pp. ISBN 0-262-19398-1,” *Neurocomputing*, vol. 35, pp. 205–206, Jan. 2000.
- [3] Q. Huynh-Thu and M. Ghanbari, “Scope of validity of PSNR in image/video quality assessment,” *Electronics Letters*, vol. 44, pp. 800–801, Feb. 2008. DOI: 10.1049/el:20080522.
- [4] S. Chandran, “Color image to grayscale image conversion,” Apr. 2010, pp. 196–199. DOI: 10.1109/ICCEA.2010.192.
- [5] A. Danielyan, V. Katkovnik, and K. Egiazarian, “BM3D frames and variational image deblurring,” *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 21, Jun. 2011. DOI: 10.1109/TIP.2011.2176954.
- [6] N. Ponomarenko, O. Ieremeiev, V. Lukin, K. Egiazarian, and M. Carli, “Modified image visual quality metrics for contrast change and mean shift accounting,” Mar. 2011, pp. 305–311.
- [7] A. Danielyan, V. Katkovnik, and K. Egiazarian, “BM3D frames and variational image deblurring,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1715–1728, 2012. DOI: 10.1109/TIP.2011.2176954.
- [8] A. Gheregá, R. Udrea, and M. Radulescu, “A Q-learning approach to decision problems in image processing,” *MMEDIA - International Conferences on Advances in Multimedia*, pp. 60–66, Jan. 2012.
- [9] J. Kober, J. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” *The International Journal of Robotics Research*, vol. 32, pp. 1238–1274, Sep. 2013. DOI: 10.1177/0278364913495721.
- [10] R. Nakamura, Y. Mitsukura, and N. Hamada, “Blind restoration of single-channel image using iterative PCA,” English, in *Proceedings - 2013 IEEE Conference on Systems, Process and Control, ICSPC 2013*, ser. Proceedings - 2013 IEEE Conference on Systems, Process and Control, ICSPC 2013, 2013 IEEE Conference on Systems, Process and Control, ICSPC 2013 ; Conference date: 13-12-2013 Through 15-12-2013, IEEE Computer Society, Jan. 2013, pp. 84–87, ISBN: 9781479922093. DOI: 10.1109/SPC.2013.6735108.
- [11] W. Pratt, “Digital image processing,” *Book*, vol. 13:978-1-48822-1670-7, 2014.

- [12] image.made-in-china, “Image.made-in-china.com,” *Website*, Sep. 2015. [Online]. Available: <https://image.made-in-china.com/202f0j00NOlTqZKBEzgU/Long-Range-Wireless-IP-Laser-Infrared-CCTV-Camera-System.jpg>.
- [13] A. Rosebrock, “Blur detection with opencv,” *Website*, Sep. 2015. [Online]. Available: <https://www.pyimagesearch.com/2015/09/07/blur-detection-with-opencv/>.
- [14] O. Anshel, N. Baram, and N. Shimkin, “Deep reinforcement learning with averaged target dqn,” Nov. 2016.
- [15] R. Bansal, G. Raj, and T. Choudhury, “Blur image detection using laplacian operator and open-cv,” in *2016 International Conference System Modeling Advancement in Research Trends (SMART)*, 2016, pp. 63–67. DOI: 10.1109/SYSMART.2016.7894491.
- [16] F. A. Fardo, V. H. Conforto, F. C. D. Oliveira, and P. S. Rodrigues, “A formal evaluation of psnr as quality measurement parameter for image segmentation algorithms,” *ArXiv*, vol. abs/1605.07116, 2016.
- [17] M. Jiu and H. Sahbi, “Laplacian deep kernel learning for image annotation,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 1551–1555. DOI: 10.1109/ICASSP.2016.7471937.
- [18] D. R. Mehra, “Estimation of the image quality under different distortions,” *International Journal Of Engineering And Computer Science*, vol. 8, Jul. 2016. DOI: 10.18535/ijecs/v5i7.20.
- [19] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” *ArXiv*, vol. abs/1602.01783, 2016.
- [20] V. Mnih, A. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” Feb. 2016.
- [21] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang, “Waterloo exploration database: New challenges for image quality assessment models,” *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 1004–1016, 2017. DOI: 10.1109/TIP.2016.2631888.
- [22] F. Pardo, A. Tavakoli, V. Levdik, and P. Kormushev, “Time limits in reinforcement learning,” Dec. 2017.
- [23] F. Pardo, V. Levdik, and P. Kormushev, *Q-map: A convolutional approach for goal-oriented reinforcement learning*, Oct. 2018.
- [24] R. Furuta, N. Inoue, and T. Yamasaki, “Pixelrl: Fully convolutional network with reinforcement learning for image processing,” *IEEE Transactions on Multimedia*, vol. PP, pp. 1–1, Dec. 2019. DOI: 10.1109/TMM.2019.2960636.
- [25] Greentec, *Learn reinforcement learning (4) - actor-critic, a2c, a3c*, May 2019. [Online]. Available: <https://greentec.github.io/reinforcement-learning-fourth-en/>.
- [26] Z. Gu, Z. Jia, and H. Choset, “Adversary a3c for robust reinforcement learning,” *ArXiv*, vol. abs/1912.00330, 2019.

- [27] R. Furuta, N. Inoue, and T. Yamasaki, “Pixelrl: Fully convolutional network with reinforcement learning for image processing,” *IEEE Transactions on Multimedia (TMM)*, vol. 22, no. 7, pp. 1704–1719, 2020.
- [28] R. Rafailov, T. Yu, A. Rajeswaran, and C. Finn, “Offline reinforcement learning from images with latent space models,” *ArXiv*, vol. abs/2012.11547, 2020.
- [29] R. Riad, A.-C. Bachoud-Lévi, F. Rudzicz, and E. Dupoux, *Identification of primary and collateral tracks in stuttered speech*, 2020. eprint: arXiv:2003.01018.
- [30] Y. Zhang, “Deep reinforcement learning with mixed convolutional network,” *ArXiv*, vol. abs/2010.00717, 2020.
- [31] *Pbs.twimg.com*. [Online]. Available: <https://pbs.twimg.com/media/Eh3jOMWUMAEwRUt.jpg>.