AI Assignment #1

Contents

Project Title

Image Defogging System (AIM LAB PROJECT)

Project Research

Introduction of Topic:

Fogging of an image can be caused by various factors and can result in a loss of detail and quality in the captured images. Defogging is a computer vision task that involves removing the fogged effect artifacts from images or videos to restore the original, sharp content. In this project proposal, we provide an image defogging application that makes use of sophisticated algorithms and methods to remove atmospheric fogging effects, converting hazy images into visually appealing and informative representations. I implemented the project from scratch.

Features of Application

1. Image Upload: Through a simple interface, the application will allow users to upload foggy photographs.
2. Image Processing: Advanced defogging algorithms will then be used on the submitted photographs to reduce the fogging effects in the image.
3. Automatic Enhancement: The automatic enhancement mode of the application will allow users to simply upload their images and let the application handle the defogging process. The algorithms used will examine the image's content, determine the amount of haze or fog present and make the necessary adjustments to improve visibility and restore clarity.
4. Manual Adjustment Tools: Users that want greater control over the image defogging process will be offered the manual adjustment tools in the application. Users will be able to fine-tune several characteristics, like the degree of fog removal, enhancement of contrast, and sharpness, using these tools, enabling customized and unique image upgrades.
5. Batch Processing: The application will feature batch processing, allowing users to upload numerous foggy photographs at the same time and have them defogged concurrently for increased user convenience and efficiency. This feature will help save time when dealing with larger collections of foggy images.
6. Comparison and Preview: A side-to-side comparison feature will be available allowing users to preview the original image alongside the defogged version. This feature will help users assess how effective the defogging process was and will help them make further adjustments if necessary.

Technical Challenges

1. Image Degradation Analysis: Robust algorithms need to be implemented/developed to analyze accurately the degree of fog in an image. Advanced computer vision algorithms and machine learning methods are needed to distinguish between fog and other image degradations like noise or motion blur.
2. Defogging Algorithm Design: It is difficult to create algorithms that effectively remove fog while retaining crucial image features. To prevent over-enhancing or the creation of artefacts, fog removal must be carefully optimized with noise suppression, contrast enhancement, and sharpness.
3. Computational Complexity: Algorithms for image defogging frequently include computationally demanding tasks. To give users a seamless experience, the application must guarantee a near real time (immediate) performance. To attain the needed performance, optimization methods, parallel computing, and hardware acceleration may be necessary.

4. Variability of Atmospheric Conditions: The application needs to handle multiple conditions, including varying levels of fog density, smog, and haze. Implementing algorithms that can generalize different conditions and adapt is an essential requirement for robust defogging performance.
5. User Interface and Experience: It is crucial to create an interface that is simple to use and intuitive to ensure that the application is usable and accessible. Real-time previews, interactive tools for manual adjustments, and a clear and understandable presentation of the many modification options were all essential factors in the development of the application.
6. Loss Estimation: The pixel-wise content loss function has to be considered in the early deep deblurring networks to measure the reconstruction error.

Project Gantt Chart

Image Defogging Using CNN

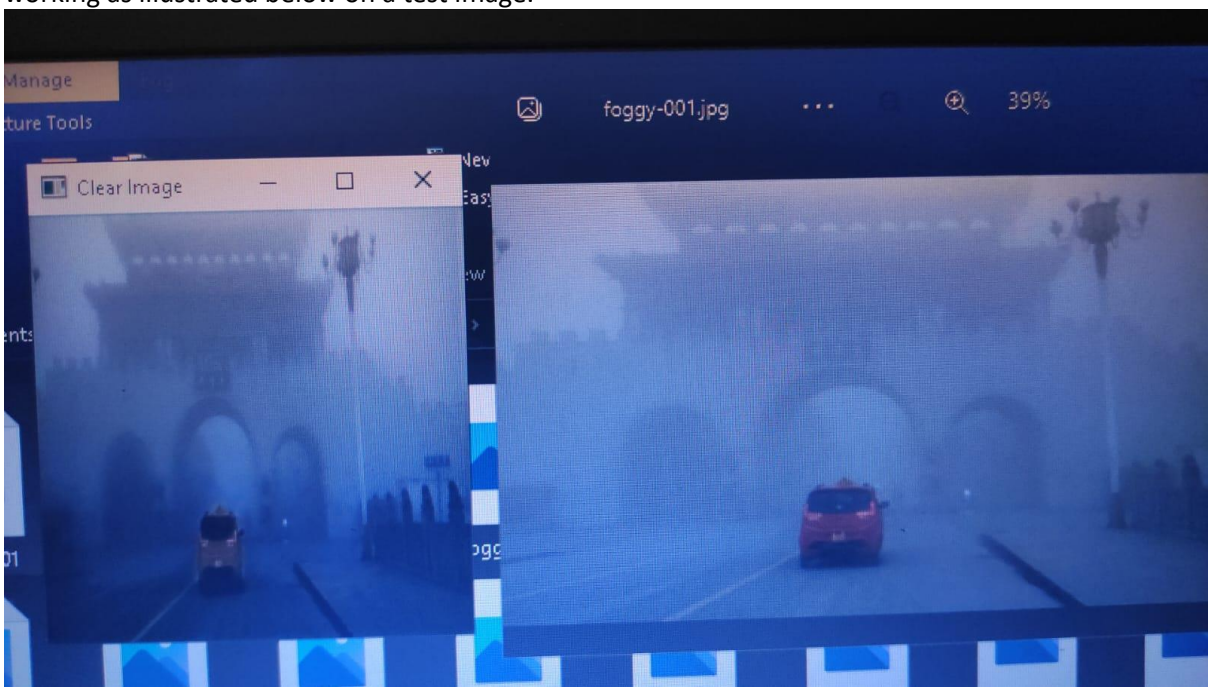
Read-only view, generated on 18 Jul 2023

	ACTIVITIES	ASSIGNEE	EH	START	DUE	%
	Image Defogging System:		-	10/Jun	26/Jul	92%
1	Project Exploration		-	10/Jun	12/Jun	100%
2	Dataset Research		-	13/Jun	05/Jul	100%
3	Front End Webpage Develo...		-	20/Jun	05/Jul	100%
4	Model Creation and Testing...		-	05/Jul	26/Jul	70%
5	Connection of Model With F...		-	05/Jul	13/Jul	100%
6	Image Dataset Preprocessi...		-	05/Jul	12/Jul	100%

Public Datasets

First Dataset: https://people.ee.ethz.ch/~csakarid/SFSU_synthetic/
The problem with this dataset was that there weren't any corresponding clear images in the dataset which was problematic for our goal, but we kept it aside for use of testing to see how effectively the images get defogged.
Second Dataset: https://www.cityscapes-dataset.com/foggydownload/
The problem with this dataset was also of only foggy images and not relevant corresponding clear images.
Third Dataset: https://github.com/Utkarsh-Deshmukh/Single-Image-Dehazing-Python
https://github.com/Infernolia/WEDGE
https://paperswithcode.com/dataset/dawn
We found these datasets but there were many images that didn't fit our goal of project so we picked the relevant images and created our own dataset of clear and corresponding foggy images. Although we were left with a small image dataset, we applied image augmentation to increase the size of the dataset.

With the help of the new dataset, I was finally able to achieve the start of the image defogging model working as illustrated below on a test image.



In the image above, the left is the predicted image and on the right is the test foggy image.

Fourth Dataset: http://perso.lcpc.fr/tarel.jean-philippe/bdd/frida.html
We were given this dataset (Frida-2 dataset not Frida-1) by our supervisor but currently this dataset is not working well with any of the several models I created.

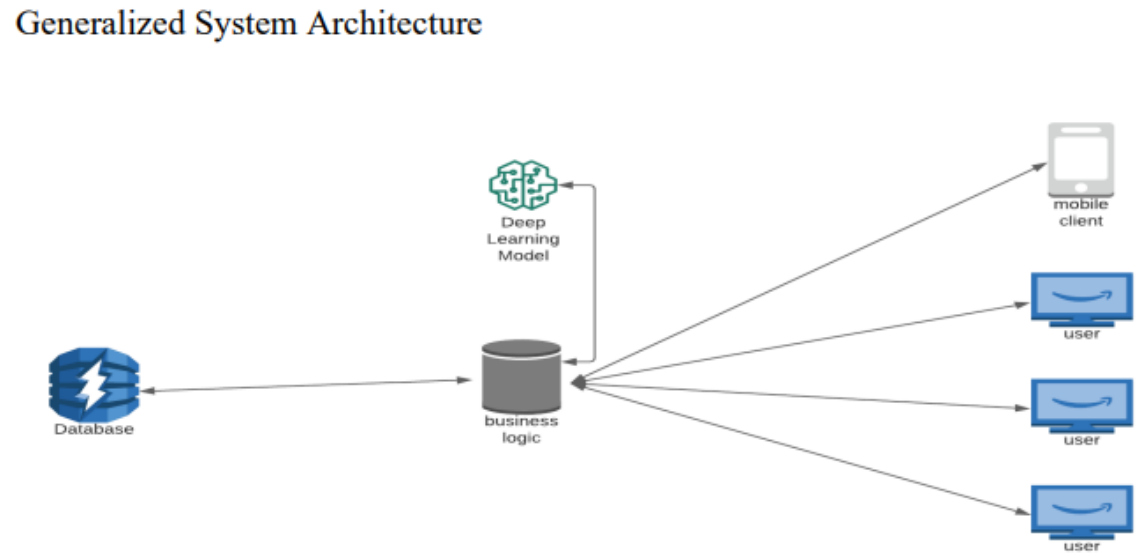
Model Research and Usage

We did a lot of research on which model to use such as FAST-R-CNN, Convnet, Dark Channel Prior(DCP), Retinex-Based models(Multi-Scale Retinex (MSR), Single-Scale Retinex (SSR), and Adaptive Manifold-based Retinex (AMSR)), DehazeNet, AOD-Net, and DehazeGAN,and CNN alongside OPENCV.

Currently I am working with CNN alongside OpenCV. Benefit of CNN is that it has a built in image feature extraction and selection based on the number of filters I apply. I used OpenCV for images storage and resizing to a fixed image height and width since varying dimensions cause a lot of problems in model training.

I used the activation function ‘relu’ for our model layers based on the research I did of which activation function would be most suitable for the task. Currently I am testing with varying model layers, batch normalization, conv2d, conv2dtranspose, epoch number, batchsize number, callback function. The callback function is to be used to interrupt the training and stop further training once a specific accuracy is reached to avoid overfitting. The TensorFlow library is vast so a lot of learning and courses I am studying online to help with better understanding of TensorFlow advanced techniques and convolutional neural networks.

Architectural Diagram



Front-End Layout (Flask and .NET Implementation)