A logo for university of canberra

Description automatically generated

***Faculty of Science and Technology***

**Assignment Coversheet**

|  |  |
| --- | --- |
| **Student ID number &**  **Student Name** | u3250716  Neil Yusuke Sutherland |
| **Unit name** | Software Technology 1 |
| **Unit number** | 4483 |
| **Unit Tutor** | Pranav |
| **Assignment name** | ST1 Capstone Project – Semester 2 2023 |
| **Due date** | 29/10/2023 |
| **Date submitted** | 29/10/2023 |

**You must keep a photocopy or electronic copy of your assignment.**

**Student declaration**

I certify that the attached assignment is my own work. Material drawn from other sources has been appropriately and fully acknowledged as to author/creator, source and other bibliographic details.

**Signature of student: NEIL SUTHERLAND Date: 29/10/2023**

Problem and Dataset Description  
The following report details the various methods of analysing a given image dataset as per the requirements of ST1’s Capstone Project. The given dataset is a public dataset on Kaggle titled “Cigarette Smoker Detection” [1].

The dataset provided for this assignment contained two classes of images – a set containing images of people smoking and a set containing people that were not smoking – coming together for a total of 3275 images. It is publicly available in a Kaggle repository and with a license specified as Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0). The images within the dataset were of varying size and in .jpg format. The images appear to have been taken from various forms of media online.

The given dataset had no description attached, so it was unclear what specific purpose or problem the dataset was attempting to solve/explore. However, based on the title and contents it can be assumed that the dataset is related to detecting the behavior of smoking in humans by comparing images of people smoking and not smoking. Many images in the non-smoking class were of people making motions/actions similar to that of smoking.

With the increasing levels of technology and advancements in image recognition software, the possibility of digitally detecting smoking behavior in videos of images through machine learning is becoming more plausible every day. After the implementation of the Tobacco Hazards Prevention Act in Taiwan, a study was proposed to use a smoking behavior detection system to detect whether a person in an image is smoking or not. This proposed system would use existing facial recognition software to detect the location of the mouth, then implement noise reduction to the image to determine the location of the cigarette (if present) [2]. The applications of such systems could help prevent smoking in prohibited areas more effectively and remove the need for smoke detectors in certain areas.

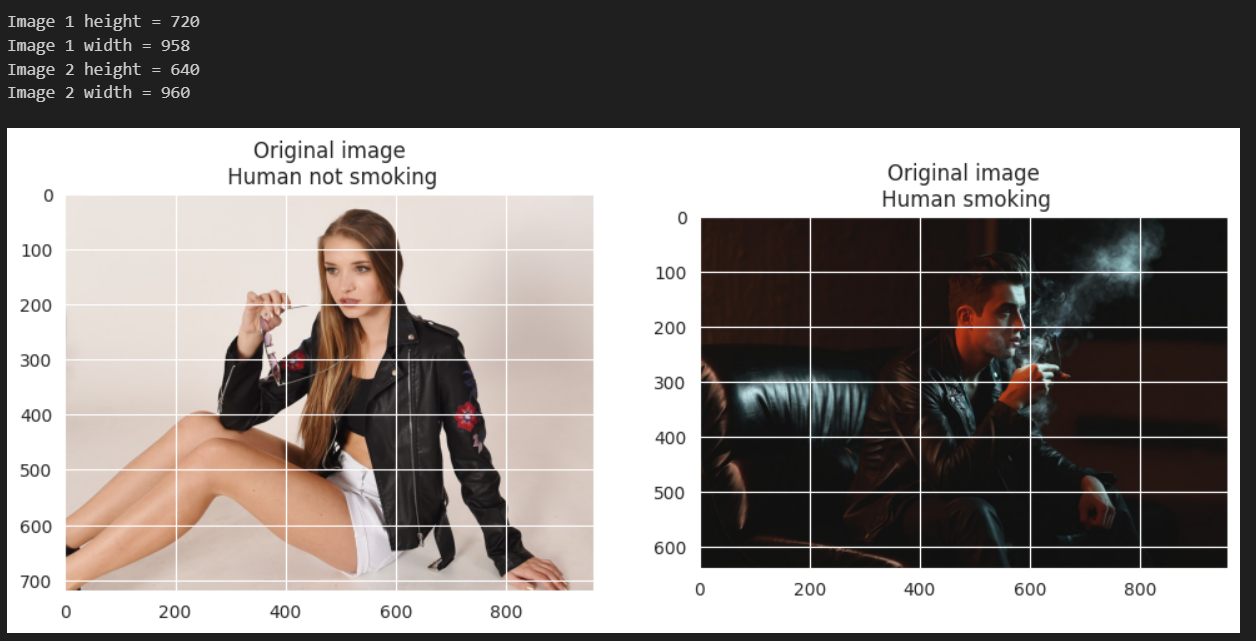
This Capstone Project Report intends to explore the possible uses of certain software in detecting smoking behavior, namely Exploratory Data Analysis (EDA) and Predictive Data Analysis (PDA). EDA will consist of various transformations to given images within the dataset and analyzing how they may be able to help with detecting smoking behavior. The PDA will use machine learning (via Teachable Machine with Google) to analyze hundreds of images from the dataset and determine if the person in the image is smoking or not.

Methodology  
**Stage 1**Exploratory Data Analysis for dataset images on Google Colab.

**Stage 2**Predictive Data Analysis for dataset images on Teachable Machine with Google.

**Stage 3**  
Implementation and Deployment of machine learning software into a GUI for real world use.

Stage 1: Exploratory Data Analysis  
EDA is the first stage of this Capstone Project and is used to obtain a thorough understanding of the data. This will be achieved by setting five questions about the images of the dataset. The various transformations that will be applied to the chosen images of the dataset will provide answers to the questions and insight into the possible applications of different software for smoking behavior detection. This report will provide the results of the EDA that was performed, and the specific code used can be found in a .ipynb folder attached to this assignment document as well as the provided GitHub repository. Relevant titles have been added above each image.

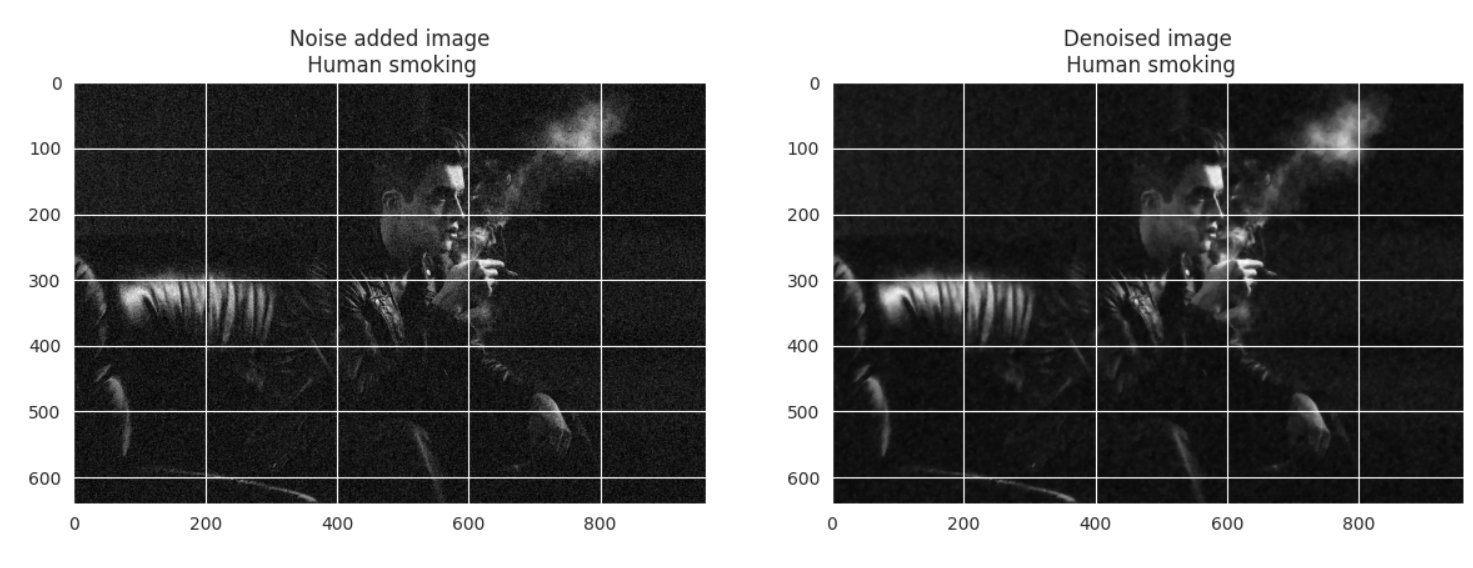
**Question 1 – How are the images displayed on a grid?**The first question and hurdle of the EDA was displaying the images via python (on Google Colab). The Matplotlib grid was used to separate the images into blocks and measure the length of the cigarette. Google Colab uses inverted colours in its displayed images by default, so some time was spent on implementing code that converts it back into the original colors using OpenCV’s convert colour. The results can be seen below:

**Question 2 – How do the images look when geometrically transformed?**The next section of the EDA involves geometrically transforming the chosen images via OpenCV in Python. The images were rotated, flipped and transposed.

A collage of images of a person

Description automatically generated

Though it is unclear how much of an effect geometrically transforming these images will have on smoking behavior detection, it is possible that these changes may impede the attempts at detection instead. As mentioned in the case study in Taiwan, existing facial recognition software is the first step in determining smoking behavior [2]. If an image of a person is vertically flipped or transposed, the facial recognition software may have a harder time detecting these faces as they are not upright. Cigarettes are often held horizontally, so transposing the cigarettes to be held vertically may further prevent the detection of smoking behavior in images.

A collage of a person posing for a picture

Description automatically generated**Question 3 – How will the images look after colour and texture tweaks?**The third stage of the EDA involved various reductions to the images’ colours and details; in other words, the images were simplified and stripped of vibrant colours and sharp edges. The first step to achieving this was grey scaling the images through Skimage to remove any colours. The grey scaled images were then noised/denoised and blurred (separately).

A person in a leather jacket

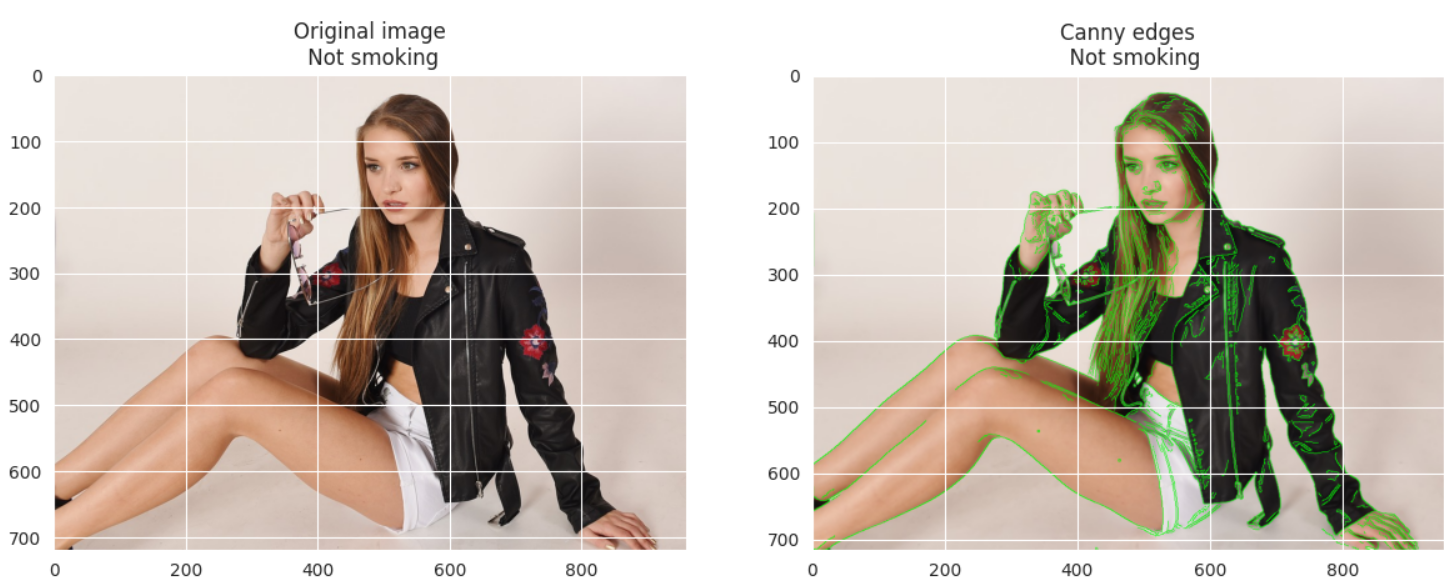
Description automatically generatedThe images with noise added produce a grainy effect and make some of the image details sharper. In the image of the man smoking, adding noise seems to have made the smoke less clear. The case study previously mentioned in the introduction proposed reducing the noise of an image in order to locate the position of the cigarette [2], so it is unlikely that denoising pictures of people smoking will assist in detecting smoking behavior. The images at the bottom have been blurred by Gaussian Blur, which uses the Gaussian Function to smooth out the randomness of image details [3]. The blur applied to the image of the man smoking appears to have slightly increased the visibility of the smoke near the top, since the white of the smoke has blended with the dark backdrop. The blurring of the smoke may assist image detection software to detect smoke itself as it increases the amount of white colour from the smoke.

A graph of a graph showing a variety of data

Description automatically generated with medium confidenceA graph of a person's body

Description automatically generated**Question 4 – How does corner detection software assist in the detection of smoking behavior?**The fourth question selected for the EDA is related to corner detection on the chosen images. Two types of corner detection software were used to highlight the edges/outlines of the images and taking the focus away from the background and large bodies of solid colour.

A collage of a person smoking a cigarette

Description automatically generatedThe first corner detecting software used was the Sobel Edge Detector, as seen in the grey images at the top. The Sobel Edge Detector works by first detecting corners that run perpendicular to the y-axis, where edges that run closer to parallel with the x-axis will have more strength while edges that are almost vertical will be faded and hard to see. The software will then detect corners that run vertically in a separate image, where any edges that run perpendicular with the x-axis will be more visible.

Cigarettes are generally held horizontally, so the Sobel Edge Detector may assist in detecting thin, horizontal lines being held by people. However, as seen in the non-smoking image, it would be easy for the detection software to mistake a similar object for a cigarette. Due to the dark background and overall similar blend of colours in the smoking image, the Sobel Edge Detector was not able to pick up any distinct edges.

The next edge detecting software used was the Canny Edge Detector which highlights any corners/edges in a thin green line, creating an outline of particular subjects within the image. Starting with the non-smoking image, the edges of the glasses frame being held was detected and outlined. This may impede attempts at detecting smoking behavior as people who are not smoking but simply making similar actions could be accused of holding a cigarette.

In the smoking image, the Canny Edge Detector can be seen outlining the smoke in the air itself. Though this may be due to the smoke’s white colour contrasting with the dark background, it is important to note that smoke is able to be detecting using corner detection. This could aid image detection software to detect smoke patterns in the air rather than cigarettes being held in a hand.

A black rectangle with a white background

Description automatically generatedA comparison of a graph

Description automatically generated**Question 5 – How does a histogram read the images after being converted to the same lighting scheme?**The final part of the EDA aims to convert both images to similar lighting states by taking only the outlines and removing the background. This will allow a histogram to treat both images fairly and compare the light levels.

An image histogram can be read using the following instructions:

* The x-axis represents a scale from 0-1.
* The left-hand side, towards the 0, represents the darkest tones (black).
* The right-hand side, towards the 1, represents the lighter tones (white).
* The higher the graph goes, the more pixels there are of that exact tone.

It was expected that when converted to similar lighting schemes, the image of the man smoking would have a graph leaning more towards the right due to the number of white particles generated by the smoke. However, as seen above the histograms for both images are very similar, with the non-smoking image having slightly lighter tones. These two images were initially chosen for their contrast in light and darkness, but it seems that changing the lighting to be the same has little effect on the detection of tones, namely the white of the smoke caused by smoking behavior.

Stage 2: Predictive Data Analysis  
The next stage of the Capstone Project is based around using machine learning to determine if an image contains smoking behavior or not. To achieve this, a model from Teachable Machine with Google was used and trained with images from the given dataset. 495 images from the “not smoking” class were chosen and 345 images from the “smoking” class to train the model. The results can be seen below.

A screenshot of a social media network

Description automatically generatedA screenshot of a computer

Description automatically generated**Image sampling:**

**Statistics:**

A screenshot of a graph

Description automatically generatedA screenshot of a cell phone

Description automatically generated

As seen in the **image sampling** section, the model was successfully trained to find the differences between smoking and non-smoking behavior by comparing a wide range of images from the dataset. Some more images (that were not used for training) were tested in the model, with similar results.

As seen in the statistics given by Google, the accuracy for the non-smoking class is 86%, while the smoking class has an accuracy of 81%. This is slightly unexpected as smoking behavior tends to be clearer and more conformed compared to non-smoking behavior, which could be a wide variety of behaviors.

Stage 3: Implementation and Deployment via Teachable Machine with Google

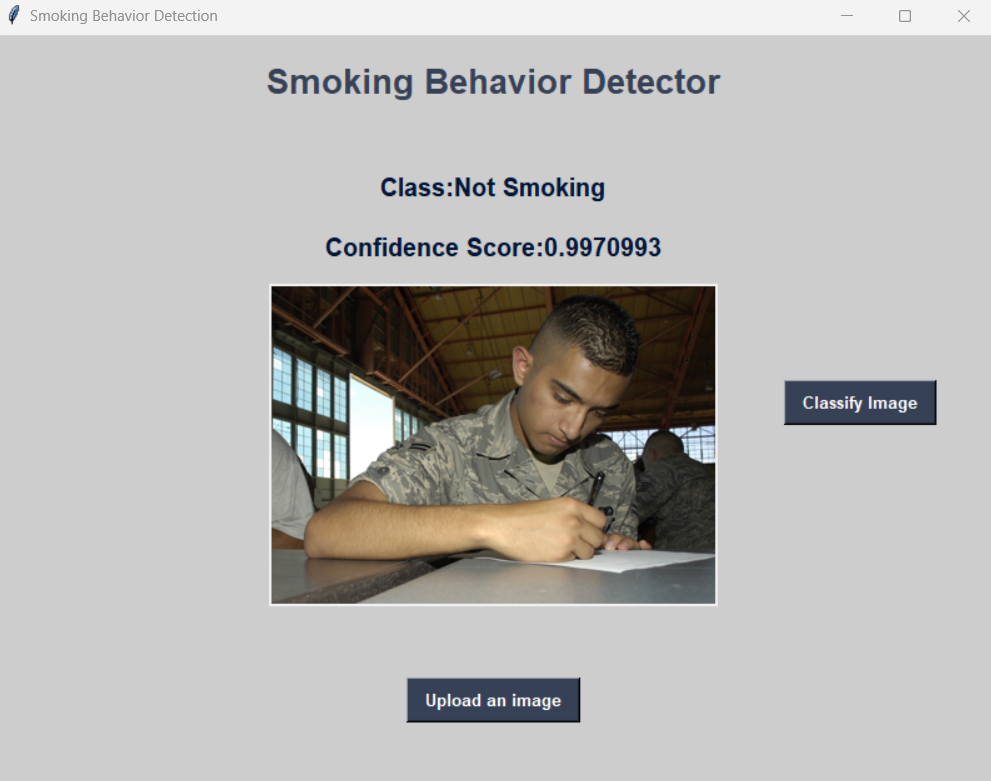
To implement the teachable model into a GUI, Tkinter was used. Taking the .h5 file received from Teachable Machine with Google, the model was implemented into a Tkinter GUI using TensorFlow. The GUI allows the user to upload an image and classify whether the image belongs to the “smoking” or “not smoking” class with the trained model. The results can be viewed in the images below.

**A screenshot of a computer

Description automatically generatedGUI starting stage (neutral)**

**A screenshot of a computer

Description automatically generated  
GUI detecting smoking behavior**

**GUI detecting non-smoking behavior**

As seen in both examples of the GUI in action, the trained model is working within the GUI and correctly assigning the given images to their correct classes, with over 99% accuracy. Further testing was carried out with similar results, marking the successful deployment of the trained model.

Conclusion  
This report presents the process and work involved in completing the Software Technology 1 Capstone Project. Given a dataset of images containing two classes – Smoking and Not-Smoking – various forms of analysis were carried out to differentiate the two. The first was exploratory data analysis, which utilized various image transformation software to test for possible applications in detecting smoking behavior. Using Teachable Machine with Google, predictive data analysis was also carried out where a machine-learning image recognition model was trained to determine which class a given image belonged to. This model was then implemented into a GUI and deployed successfully, correctly detecting whether or not an image displayed smoking behavior.

Based on the results of this report, there may be more ways to detect smoking behavior in images and videos. For example, the EDA showed that the Canny Edge Detector is able to find the outlines of smoke and the deployed model in the GUI was able to differentiate smoking behavior and non-smoking behavior with over 99% confidence. A combination of these two methods may improve current smoking behavior detection and help prevent smoking in restricted areas.

Reference List

1. <https://www.kaggle.com/datasets/vitaminc/cigarette-smoker-detection>
2. W. -C. Wu and C. -Y. Chen, "Detection System of Smoking Behavior Based on Face Analysis," 2011 Fifth International Conference on Genetic and Evolutionary Computing, Kitakyushu, Japan, 2011, pp. 184-187, doi: 10.1109/ICGEC.2011.51.
3. <https://datacarpentry.org/image-processing/06-blurring.html>

Assignment requirements

*Relevant files have been included in the assignment submission, as well as any completed challenge questions. Logbook can be found below.*

ST1 Capstone Project Logbook (Appendix)

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Planned Activities** | **Tasks Completed** | **Issues Faced** |
| 10  (02/10/2023) | Unfortunately, the first week I’m supposed to be keeping a logbook, my ST1 tutorial is placed on a public holiday and not running. I was unable to find another tutorial to attend due to other commitments and therefore was unable to check up on my progress. However, I plan to finish my EDA by next week. | I have completed the five questions required for my Exploratory Data Analysis and am beginning to transform my given images in Google Colab accordingly. | It was difficult to adapt to the new style of coding (in blocks) and I initially had trouble getting the images to display normally as Google Colab inverts the colours of images by default. This required writing some extra code and converting the colours back to normal. |
| 11  (09/10/2023) | This week I was able to check in with my tutor Pranav and ask some questions, though there wasn’t much time since the other students also had a lot to ask. This week I plan to begin my PDA and play around with different teachable models. I haven’t decided which method to use yet, but Teachable Machine with Google looks easier to understand. | My EDA has been completed and I am beginning to form ideas on how the different images transformations are relevant to detecting smoking behavior. Some are much more relevant than others. I have saved the Google Colab file to my computer. | Many of the transformations I made as part of my EDA questions held little to no relevance to detecting smoking behavior, and I wondered if they should be included in the report. I also had trouble processing the image randomizer with the GPU. |
| 12  (16/10/2023) | I was unable to check on my progress with my tutor this week as he was assessing week 12 presentations/interviews. I stuck around to watch the presentations of my fellow students, then left. I hope to begin the implementation of my trained model into a GUI this week, but it looks slightly complicated. I hope to also compile my findings from EDA and PDA into my report. | My PDA is finished, and since it took quicker than I expected I have also begun writing my Capstone Project report. The EDA section has been completed and I have written what I can in the introduction about the (assumed) problem at hand. | Some of the EDA questions I had asked were irrelevant so, it was difficult to write about them in the report. The dataset I had been given also had no description, so I did not know what the problem description was or what the dataset was trying to solve. I assumed it had to do with detecting smoking behavior and found a relevant case study. |
| 13  (23/10/2023) | I presented my Capstone Project presentation/interview to my tutor today. He didn’t say much, other than the fact that I didn’t need a case study. I have yet to begin implementing my trained model into a GUI as the report has taken priority, but I plan to finish it within the next couple of days. | My report is virtually complete, sans the implementation of the model into the GUI. I have been working on some overdue challenge questions as well. I plan to create the GUI from Tkinter. | The implementation of the GUI is still a bit confusing as I am not sure how to bring the trained model from Teachable Machine with Google into Tkinter. I plan to continue working at it. |