This is The Title of Your Special Problem

A Special Problem
Presented to
the Faculty of the Division of Physical Sciences and Mathematics
College of Arts and Sciences
University of the Philippines Visayas
Miag-ao, Iloilo

In Partial Fulfillment
of the Requirements for the Degree of
Bachelor of Science in Computer Science by

LASTNAMEA, FirstName1
LASTNAMEB, FirstName2
LASTNAMEZ, FirstName3

Firstname LASTNAME
Adviser

December 23, 2022

Abstract

From 150 to 200 words of short, direct and complete sentences, the abstract should be informative enough to serve as a substitute for reading the entire SP document itself. It states the rationale and the objectives of the research. In the final Special Problem document (i.e., the document you'll submit for your final defense), the abstract should also contain a description of your research results, findings, and contribution(s).

Suggested keywords based on ACM Computing Classification system can be found at `https://dl.acm.org/ccs/ccs_flat.cfm`

Keywords: Keyword 1, keyword 2, keyword 3, keyword 4, etc.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview of the Current State of Technology

This section gives the reader an overview of the specific technology or field in the international or local setting. The information regarding the technology or field should be contemporary and not based on outdated sources. Discussion must not be too technical or too detailed.

This section ends with a discussion on the problem/s faced by or that still exist in the specific technology or field (e.g., limitations of existing software or algorithms). The problem statement would lead to the research objectives.

It is easy to include a figure in JPG or PNG format as shown in the following example. Make sure that you explain what the figure is all about, and that you refer to your figure. For example, Figure 1.1 shows a graph of the performance of Disney stock from the 1980s to 2012.

Some notes on citing references. When using APA format, the author-date method of citation is followed. This means that the author's last name and the year of publication for the source should appear in the text, and a complete reference should appear in the reference list.

Here are some examples on how to do the referencing (note author's name and years are different from commented examples). For APA citation details, refer to `http://www.ctan.org/tex-archive/biblio/bibtex/contrib/apacite/`.

- Kartch (2000) compared reaction times...

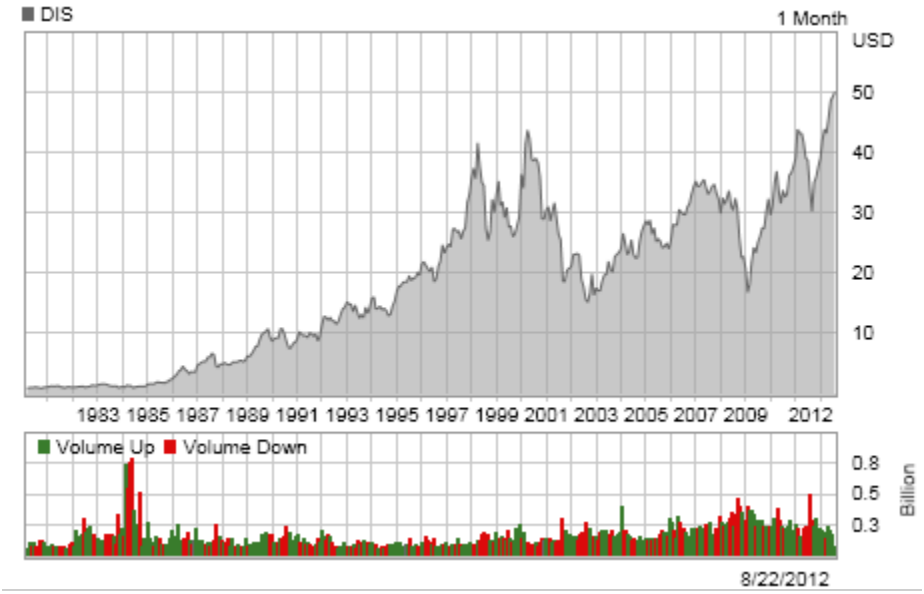

Figure 1.1: This is the figure's caption – Disney stock chart. Captions should fully describe the figure in a concise manner such that there is not need to refer to the text when figuring out the graphic.

- In a recent study of reaction times (Kartch, 2000)...
- In 2000, Kartch compared reaction times...
- Fedkiw et al. (2001) compared reaction times...
- In a recent study of reaction times (Fedkiw et al., 2001)...
- In 2001, Fedkiw et al., compared reaction times...

The following are references from journal articles (Park, Linsen, Kreylos, Owens, & Hamann, 2006; Pellacini et al., 2005; Sako & Fujimura, 2000). Here's an MS thesis document (Yee, 2000), and this is from PhD dissertation (Kartch, 2000). For a book, reference is given as (Parke & Waters, 1996). Proceedings from a conference samples are (Jobson, Rahman, & Woodell, 1995; Fedkiw et al., 2001; Levoy et al., 2000). The sample bibliography file named **myreferences.bib** is from the SIGGRAPH LaTeX template. You can use a text editor to view the contents of the bib file. It is your task to create your own bibliography file. For those who downloaded papers from ACM or IEEE sites, there is a BibTeX link that you can click; thereafter, you just simply need to copy and paste the BibTeX entry into your own bibliography file.

The following shows how to include a program source code (or algorithm). The verbatim environment, as the name suggests, outputs text (including white spaces) as is...

```
#include <stdio.h>
main()
{
     printf("Hello world!\n");
}
```

1.2 Problem Statement

DO NOT FORGET to write the statement of the research problem here, i.e., before the Research Objectives.

A problem statement is your research problem written explicitly. The problem statement should do four things:

1. Specify and describe the problem (with appropriate citations)
2. Provide evidence of the problem's existence
3. Explain the consequences of NOT solving the problem
4. Identify what is not known about the problem that should be known.

1.3 Research Objectives

1.3.1 General Objective

This subsection states the over–all goal that must be achieved to answer the problem. Address the following: Given your research challenge or opportunity, how do you intend to solve it? What is the output of your research?

1.3.2 Specific Objectives

This subsection is an elaboration of the general objective. It states the specific steps that must be undertaken to accomplish the general objective. These objectives must be **S**pecific, **M**easurable, **A**ttainable, **R**ealistic, **T**ime-bounded. A specific objective start with "to <verb>" for example: to design/survey/review/-analyze.

Studying a particular programming language or development tool (e.g., to study Windows/Object-Oriented/Graphics/C++ programming) to accomplish the general objective is inherent in all thesis and, therefore, must not be included here.

1. To review related literature, compare and contrast existing algorithms (on what problem?);
2. To develop a new algorithm (for what purpose?)
3. To analyze the algorithm (based on what criteria?)

1.4 Scope and Limitations of the Research

This section discusses the boundaries (with respect to the objectives) of the research and the constraints within which the research will be developed.

1.5 Significance of the Research

This section explains why research must be done in this area. It rationalizes the objective of the research with that of the stated problem. Avoid including sentences such as "This research will be beneficial to the proponent/department/college" as this is already an inherent requirement of all BSCS majors. Focus on the research's contribution to the Computer Science field.

The following are guide questions that may help your formulate the significance of your research.

- What is the relevance of your work to the computer science community?

  - What will be your technical contributions, in terms of algorithms, or approaches, or new domain?
  - What is your value-added compared to existing systems?

- What will be your contributions to society in general?
  - Who will benefit from your system?
  - Who are your target users and how will this system benefit them?

Chapter 2

Review of Related Literature

This chapter discusses the features, capabilities, and limitations of existing research, algorithms, or software that are related/similar to the Special Problem.

The reviewed works and software must be arranged either in chronological order, or by area (from general to specific). Observe a consistent format when presenting each of the reviewed works. This must be selected in consultation with the adviser.

DO NOT FORGET to cite your references.

2.1 Air Quality Monitoring Systems

Air quality monitoring systems are systems that collects data to record and or or analyze atmospheric emission levels. There are various systems for air quality monitoring. Zoogman et al. (2016) showcased in a journal the use of satellite imagery for large-scale air quality monitoring. They call this instrument TEMPO (Tropospheric Emissions: Monitoring of Pollution). It is an instrument that collects data on tropospheric emissions such as NO2, SO2, H2CO, Methane, etc from a satellite in a geostationary orbit. This system is wide-range and precise, however, access to the equipment is limited. A more accessible air monitoring system was made byy Zheng et al. (2016) using several sensors. This system makes use of LPWA (low power wide area) to give it a wider coverage compared to the IoT (Internet-of-Things) and the air quality data can be accessed through a mobile application. These systems make use of dedicated sensors to collect emission data whereas this project will make use of computer vision and machine learning.

2.2 Air Pollution from Vehicles

The Philippines currently has a problem with air pollution. According to Tentengco et al. (2022), the Philippines' PM2·5 concentrations in urban areas exceed the WHO guideline value. They further state that the Philippines' PM2·5 levels reaches 58·4 ug/m3 in traffic sites of Metro Manila during the dry season. Though there could be different sources of air pollution, 65 percent of the air pollutants come from mobile sources such as: cars, motorcycles, trucks, and buses (DENR Philippines, 2018).

Furthermore, CO2, a component of greenhouse gasses, totaled to "30 million tons and 56 thousand tons of particulate matter" (Herbert & Sudhir, 2009) in the Philippines and the transport sector contributed to 38 percent of fuel combustion back in 2000. The authors have noted that the motorized vehicle count would double by 2020. The increase of motorized vehicles also means an increase in its air pollution contribution.

2.3 Vehicle Tracking

Vehicle tracking can be used for identifying the information of the vehicle, such as its brand and model. A recent paper by Saravi and Edirisinghe (2019) presents a Vehicle Make and Model Recognition (VMMR) System that accepts a video feed and returns the make and model of the vehicles detected on the feed. This system tracks the vehicle's license plate followed by selecting the region of interest above the plate–the vehicle. As the vehicle moves along the camera, its license plate is also detected across multiple frames while motion segmentation was used to keep track of the static area above the license plate.

Aside from finding the vehicle's plate number, another process to detect and track the vehicle would be through background subtraction. Background subtraction, according to Huang BJ. et al.(2017, as cited in Manzoor, 2018), is used to extract the moving objects and then filter the unwanted images through image processing tools.

2.4 Application of YOLO

From the YOLOv5 Documentation (https://docs.ultralytics.com/) website: "YOLO, an acronym for 'You only look once', is an object detection algorithm that divides

images into a grid system. Each cell in the grid is responsible for detecting objects within itself." This is a useful tool for identifying objects in an image or video.

One application of this algorithm was done by Yan et al. (2021) for an apply picking robot. YOLOv5 was used to identify apples, however the algorithm cannot detect apples that are safe to pick and those that are not. This may cause the picking arm of the robot to break if it tries to grasp an apple that is occluded by a solid object. They solved this problem by improving on the modules used for the algorithm. This is not a problem for this project as it only counts the number of vehicles without interacting with them.

In a study done by Zhou et al. (2021), they applied YOLOv5 algorithm to detect safety helmets on workers. The algorithm had an average detection speed of 110 fps in real-time. With a 94.7% effectiveness (The model was trained and tested using 6045 data sets) the algorithm proved to be viable for real-time detection.

Chapter 3

Research Methodology

This chapter lists and discusses the specific steps and activities that will be performed to accomplish the project. The discussion covers the activities from pre-proposal to Final SP Writing.

3.1 Research Activities

Research activities include inquiry, survey, research, brainstorming, canvassing, consultation, review, interview, observe, experiment, design, test, document, etc. Be sure that for each method, process, or algorithm used, there is a justification why that method was chosen. The methodology also includes the following information:

- who is responsible for the task
- the resource person to be contacted
- what will be done
- when and how long will the activity be done
- where will it be done
- why should be activity be done

3.2 Calendar of Activities

A Gantt chart showing the schedule of the activities should be included as a table. For example:

Table 3.1 shows a Gantt chart of the activities. Each bullet represents approximately one week worth of activity.

Table 3.1: Timetable of Activities

Activities (2009)	Jan	Feb	Mar	Apr	May	Jun	Jul
Study on Prerequisite Knowledge			••	••••			
Review of Existing Racing Strategies	••	••••	••••	••••			
Identification of Best Features				••••	••		
Development of Racing Strategies				••	••••	••	
Simulation of Racing Strategies				••	••••	•••	
Analysis and Interpretation of the Results					••••	••••	•
Documentation	••	••••	••••	••••	••••	••••	••

Chapter 4

Preliminary Results/System Prototype

4.1 Training the Data Set

Since the Vehicle Traffic data set was already preprocessed, it only needs to be trained. This as done by running the following code in Google Colab:

```
!git clone https://github.com/ultralytics/yolov5
%cd yolov5
%pip install -qr requirements.txt

import torch
import utils
display = utils.notebook_init()
```

This prototype training will only train for 100 epochs.

```
!python train.py --batch 16 --epochs 100 --data
   /content/drive/MyDrive/College/SP/data/data.yaml --weights
   yolov5s.pt --cache
```

4.2 Results

Figure 4.1 shows the statistics of how the data set performed during training. Notice that as the training progress the loss values drops. This is the desired behavior as it shows that the training is making less mistakes as training continues. Although, This will improve with further training.

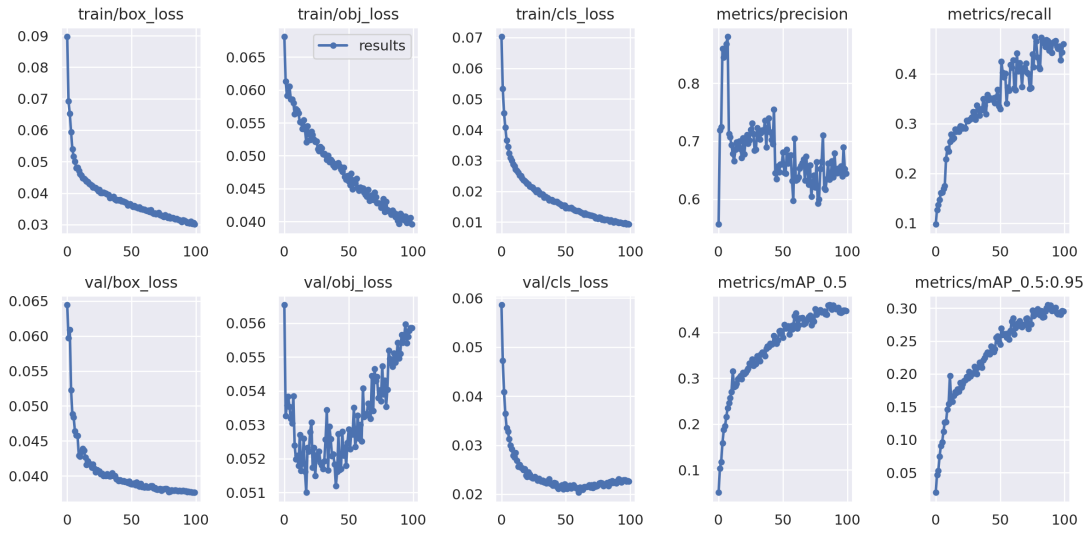

Figure 4.1: Statistics of the prototype training

The precision is expected to increase however in the results displayed the opposite. This is not desirable as it means that the model is getting less precise as training goes on. Although, the model that will be used is the best performing one.

4.3 Object Detection

The trained weights obtained by training was used in prerecorded videos to determine if the weights are trained successfully.

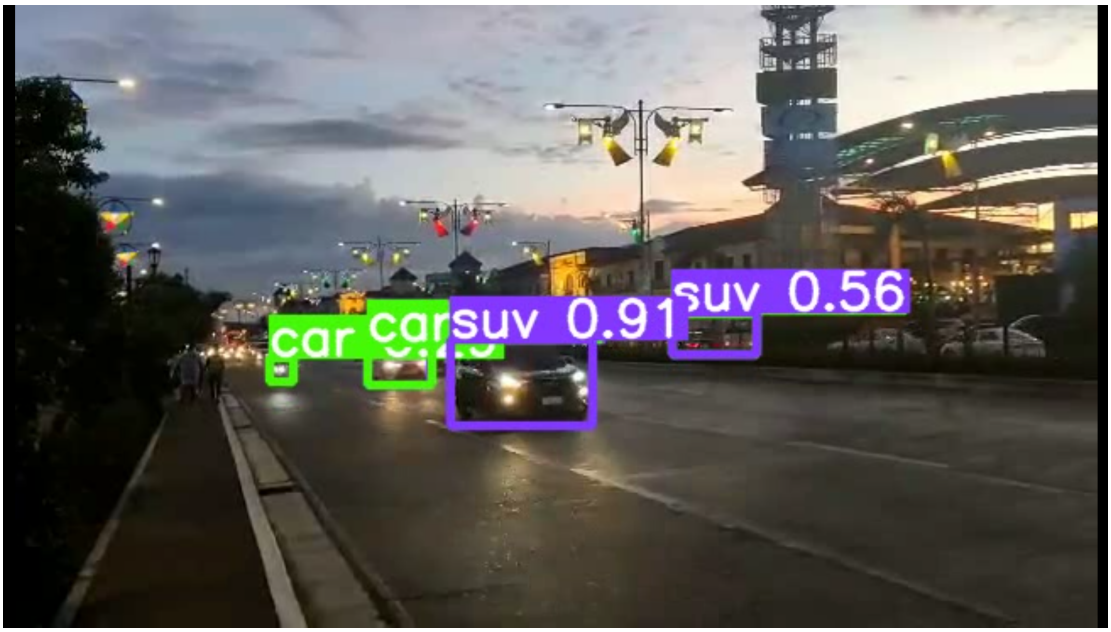

Figure 4.2: Object Detection Prototype used for Traffic recorded in street view

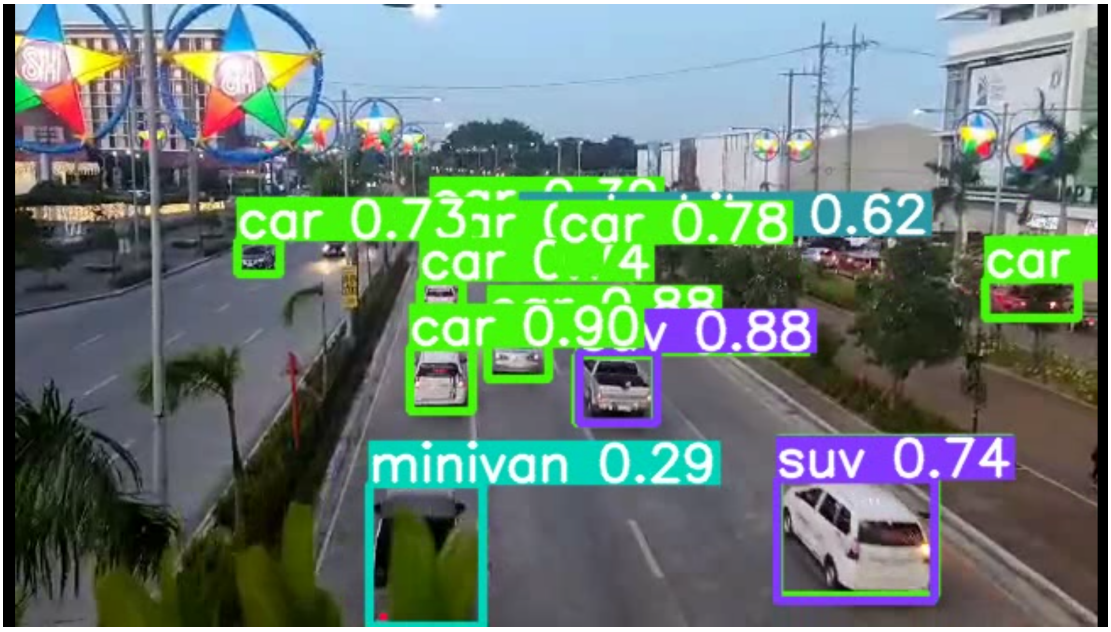

Figure 4.3: Object Detection Prototype used for Traffic recorded in bird's eye view

Appendix A

Appendix

Appendix B

Resource Persons

Dr. Firstname1 Lastname1
Adviser
Affiliation1
emailaddr@domain.com

Mr. Firstname2 Lastname2
Role2
Affiliation2
emailaddr2@domain.com

Ms. Firstname3 Lastname3
Role3
Affiliation3
emailaddr3@domain.net

References

Fedkiw, R., Stam, J., & Jensen, H. W. (2001). Visual simulation of smoke. In E. Fiume (Ed.), *Proceedings of siggraph 2001* (pp. 15–22). ACM Press / ACM SIGGRAPH.

Jobson, D. J., Rahman, Z., & Woodell, G. A. (1995). Retinex image processing: Improved fidelity to direct visual observation. In *Proceedings of the is&t fourth color imaging conference: Color science, systems, and applications* (Vol. 4, pp. 124–125).

Kartch, D. (2000). *Efficient rendering and compression for full-parallax computer-generated holographic stereograms* (Unpublished doctoral dissertation). Cornell University.

Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., ... Fulk, D. (2000). The digital michelangelo project. In K. Akeley (Ed.), *Proceedings of siggraph 2000* (pp. 131–144). New York: ACM Press / ACM SIGGRAPH.

Park, S. W., Linsen, L., Kreylos, O., Owens, J. D., & Hamann, B. (2006, March/April). Discrete sibson interpolation. *IEEE Transactions on Visualization and Computer Graphics*, *12*(2), 243–253.

Parke, F. I., & Waters, K. (1996). *Computer facial animation*. A. K. Peters.

Pellacini, F., Vidimče, K., Lefohn, A., Mohr, A., Leone, M., & Warren, J. (2005, August). Lpics: a hybrid hardware-accelerated relighting engine for computer cinematography. *ACM Transactions on Graphics*, *24*(3), 464–470.

Sako, Y., & Fujimura, K. (2000). Shape similarity by homotropic deformation. *The Visual Computer*, *16*(1), 47–61.

Yee, Y. L. H. (2000). *Spatiotemporal sensistivity and visual attention for efficient rendering of dynamic environments* (Unpublished master's thesis). Cornell University.