Traffic Object/Lane Detection

Project Proposal

OLD482

Khai Nguyen

Viet Nguyen

Pavan Poladi

Jason Bernal

Department of Computer Science

Texas A&M University

09/26/2022

Table of Contents

1 Executive summary

When many think of lane and object detection, autonomous vehicles quickly come to mind. This makes sense as detection models are crucial for the success of autonomous vehicles. However, the applications for these detection models do not stop there. Many different types of robotics can use these detection models to their advantage and achieve very impressive and unique results. Some examples include everyday gas or electric vehicles, small robots delivering packages and food throughout universities, healthcare robots used to assist patients and doctors, and many more. This is why the Robot Operating System (ROS) framework has received high praise for their work in this field. ROS is a set of software libraries and tools for building robot applications. Our team intends to assist the ROS group by helping them solve one of their problems. Currently, a separate lane detection framework and separate object detection framework are being ran by everyday ROS users. This is problematic as it decreases efficiency and usability, creates room for possible discrepancies between results from the lane detection model versus the object detection model, and also prevents future growth for ROS by not being able to capture the benefits from using shared results and data from a joint detection model. We as a team, need to create a clear and flexible pipeline allowing ROS to integrate an existing lane and object detection framework so users of ROS that are interested in this field can have a more concise and simple experience when implementing these features. This is a clear need we wish to meet and we hope that the ROS team can incorporate this integration in the near future. Our goal to solve this need is to replace ROS' current system of using two separate detection models by creating and integrating a pipeline for ROS using an existing combined lane and object detection framework. We will first set up our ROS environments and begin exploring the ROS framework so that we can become well acquainted with how the system is organized and how all the nodes interact with one another. We will then begin testing ROS' current lane detection and object detection systems and measure their efficiency, accuracy, intuitiveness, and overall simplicity. Our third objective is to try out the existing combined lane and object detection framework that ROS is not currently using and measure the same metrics to make comparisons between the two systems and evaluate how compatible ROS' framework and this new detection model are together. Our final objective is to develop a pipeline using the ROS framework and the existing combined lane and object detection framework to allow a user to take an image as an input and return an image with bounding boxes and classification results, all within the ROS framework. We hope to achieve great results and benefits for this project, and thank the ROS team for allowing us to take a shot at helping their cause and mission. We have a few specific results that we have created with the various constraints of our project in mind. Firstly, and most importantly, we hope to give the ROS team insight on how well other existing frameworks that combine lane and object detection perform compared to ROS' usage of their separate detection models. This itself would be a success if achieved, as it will provide ROS with great data and will help their cause greatly by knowing how much focus they should draw their attention to within this problem. Another result we hope to achieve is helping ROS understand how well their framework integrates with other existing frameworks. This will help the team decide whether they should make any fundamental changes to how their system is organized. Furthermore, we would also like to take this experience as a learning opportunity to see how interested we are in this subject matter, how much we can learn about machine learning and detection models, and how quickly we can onboard onto a project of large scale like this. Lastly, we would like to see how effectively we as a team can work together to achieve our objectives and make an impact in this space for our final capstone project.

2 Introduction

2.1 Needs statement

This ROS framework currently contains a separate lane detection framework and a separate object detection framework. This creates a decrease in efficiency, usability, and simplicity as one would need to run their models through both detection frameworks separately. By having these be two separate detection models, this also leaves more room for discrepancies between the two models as well as preventing future growth from not being able to combine shared results and data from a joint detection model to get unique perspectives. We as a team need to make ROS' use of object and lane detection more efficient, accurate, and intuitive. Ideally, we need to essentially replace the current system which has a separate lane detection framework and object detection framework, so that ROS can reek all the benefits of having a combined lane and object detection framework, including being able to make future improvements more easily.

2.2 Goal and objectives

Our goal is to create a clear and flexible pipeline allowing ROS to integrate an existing combined lane and object detection framework so users of ROS that are interested in this field can have a more smooth, simple, and confident experience when implementing their work. Ideally, we want to create a system that is more efficient, more accurate, and more flexible than the current ROS pipeline. Our first objective is to first fully understand the ROS framework so that we can truly understand how to take advantage of key features of ROS to help reach our goal. Our second objective is to begin testing ROS' current lane detection and object detection systems to see how efficient they are, how accurate they are, how intuitive and flexible they are, and how easy it is for a new user to begin integrating these systems. Our third objective is to try out the existing combined lane and object detection framework that ROS is not currently using to see how efficient, accurate, and intuitive it is compared to ROS' separate lane and object detection systems. This step will also allow us to evaluate the pros and cons of integrating this system as well as how compatible it is with ROS' framework. Our final objective is to develop a pipeline using the ROS framework and the existing combined lane and object detection framework to allow a user to take an image as an input and return an image with bounding boxes and classification results, all within the ROS framework. It is important to note that we are determined to achieve out objectives with zero additional spending and only using our own physical computing machines/laptops to run the complex algorithms and large data sets to emulate how a typical new ROS user would eventually use the integrated pipeline.

2.3 Design constraints and feasibility

There are many constraints we have to consider when evaluating the feasibility of our project objectives. One technical constraint is that our team does not have great experience in the topic of machine learning and graph architectures beyond the knowledge we have gained from a basic data structures and algorithms course. This is necessary to truly understand the ROS framework as well as the existing lane and object detection framework. The more familiar we are with these concepts, the more likely we will be able to make a more efficient and intuitive pipeline that will take full advantage of ROS' capabilities. Another technical constraint is the lack of knowledge on robotics within our team. Having a good knowledge of robotics would allow our team to consider other possibilities for our pipeline and have a better understanding of how it can actually be used in various scenarios involving robotics beyond just autonomous and non-autonomous vehicles. Another technical constraint is that ROS has been being developed for years and we simply have never used the framework. We would be in a much better position and have much higher chances of meeting our objectives if we had prior experience using ROS or similar frameworks. This leads us to a major temporal constraint. We only have about 10 weeks starting from the day of our project proposal presentation to meet our objectives. With more time, we would be able to create a much better pipeline and system which would be far more likely to actually be integrated into the ROS framework in the future. We also have a major physical constraint being that we are only a team of 4 computer scientist undergraduates. Logically speaking, if we had a larger team or members that were more experienced, we would simply be able to achieve much better results and have higher chances of meeting our objectives. Another physical constraint is the performance capabilities of our computers. The performance and amount of testing we are able to achieve is highly determined on how capable our systems are to run these complex algorithms on large data sets. Lastly, we have to consider economic constraints. Many of the existing lane and object detection frameworks out there are free for the public to use, but perhaps there are paid frameworks which could be far better for the purpose of reaching our goals. We want to achieve objectives with zero additional spending, which could limit the possibilities of the overall framework achieving the greatest success.

3 Literature and technical survey

Until now, there has been a lot of research related to computer vision by many researchers approaching lane detection and object detecting. The milestones of lane detection and object detection such as Fast RNN, and YOLOv3. Here are some existing products and researches that will be used in this project.

Project 1: Study of object detection based on Faster R-CNN [5]

The backbone of this project is that it combined the RPN network and the Fast R-CNN network is one of the best ways to object detection of R-CNN series based on deep learning. This paper also claimed that the overall performance is greatly improved. In fact, according to a recent ranking, it ranks 10 among the similar networks.. In the result, the training model and classifications result of mean average precision (mAP) is obtained. In the conclusion, they conclude that the mean average accuracy results are not high. However, experimental results show that its effectiveness comes from the convolutional layers and region proposal module.

Project 2: EfficientDet: Scalable and Efficient Object Detection [6]

The main concern of this paper will be to explain the model's efficiency. This concert has become increasingly important in the computer vision industry. In the paper, they also propose a weighted bi-directional feature pyramid network (BiFPN). For more information, the BiFPN allows simple and fast multiscale feature fusion. This will solve the problem of backbone structure when they are dealing with depth, width, and resolution feature maps from the backbone. This paper shows the optimization and better backbones. As the result, based on their optimizations, they developed a new family of detections, named EfficientDet which has better results in terms of accuracy and efficiency than the previous model.

Project 3: Flexible lane detection using CNNs [7]

Lane detection has become a critical feature in the autonomous driving research field. There are two ways of achieving this result. First, we apply the traditional machine learning methods, and the other way is hot deep planning which is very popular in recent years. As the environmental factors are complex, therefore when applying the traditional machine learning model, cost greatly in computations. In this paper, the authors proposed a method that uses a cascade convolutional neural network structure for end-to-end lane detection. In this model, they also explain the encoders and decoders which help them achieve the result. The lane boundary is segmented first, then following the lane classification.

Project 4: YOLOPv2: Better, Faster, Stronger for Panoptic Driving Perception [8]

Over the last 10 years, multi-tasking in automobiles has achieved a huge demand. When designing a network for autonomous vehicles, this feature should be included and implemented in almost real-time. To achieve this achievement, the authors have succeeded in creating state-of-the-art performance with more accuracy and speed on the BDD100K dataset. The most successful that they have achieved is that the inference time is reduced by half compared to the previous model. The network of YOLOPv2 reduces the cost greatly thanks to its network structure. This is the model that is ranked higher than the model that we are planning to use.

Project 5: A Clustering Method Based on K-Means Algorithm [9]

The main purpose of this paper is to combine the traditional K-Means algorithm to propose an improved K-Means clustering algorithm and the largest minimum distance algorithm. This will solve the problem of making up for the shortcomings of the traditional K-Means algorithm to determine the initial focal point. This model will keep the high efficiency and have a higher speed of convergence effectively by improving the way of selecting the initial cluster focal point.

4 Proposed work

4.1 Evaluation of alternative solutions

- ROS2 instead of ROS1
- YOLOPv2
- Last Resort
- From Scratch
- More Power

It has been suggested that we use ROS2 over ROS1 since it may offer alternative performance that could do better. It is valid as a stretch goal since there are many similarities between the two and in the best case scenario, it will only have a couple of modifications. Some research reveals that there are ROS1 and ROS2 bridges so that ROS1 and ROS2 nodes can communicate with each other. In terms of future proofing, it seems like ROS2 is better to learn if we are considering personal education and growth since it will be more widely used as time goes on.

An alternative solution is that we use YOLOPv2 instead of HybridNets. However, this is very unknown in the amount of time it would take to implement this since we are going off the suggested idea. It is also very relatively new, seeing as it was released in August 2022, with what seems to be much less instructions on their Github. It being new could be a con, or it could be a positive, but there might be unforeseen problems.

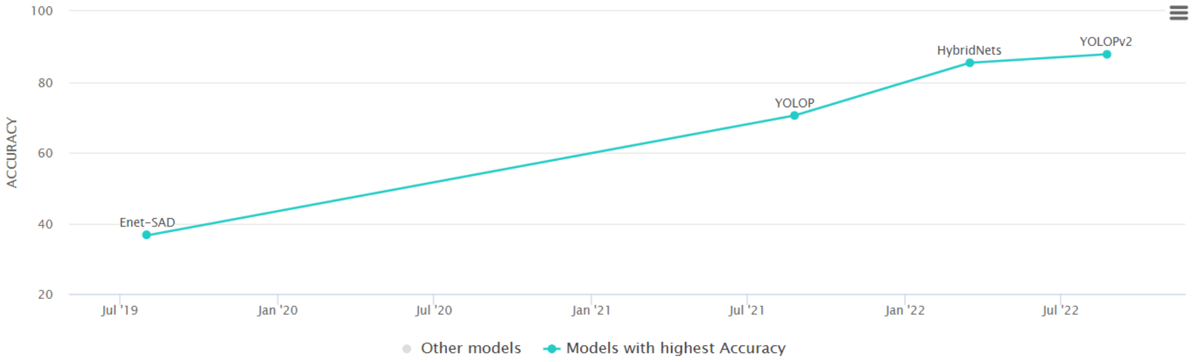


Figure 1: Comparison of Frameworks

Interestingly enough, they have a chart with highest accuracy where they boast being the best by a small margin compared to HybridNets, which is the proposed framework that we will use for our current project. At the moment, on their paper, they don't seem to have anything that would suggest it would be more lightweight, but perhaps this can be an avenue for another team to pursue in the future if HybridNets doesn't perform as efficiently as desired.

As a final resort if our system/process is a lot slower, we will fall back on the original solution having two individual modules for traffic light negotiation and lane detection but this is not ideal since we have been told that it is only powerful enough to have one running but not the other since there is a limited amount of power. An advantage to this is more time maybe can be put into optimizing individual modules. At the very least, we have knocked a possible avenue of solution for the Autodrive Challenge as a whole which benefits the future teams, not just by process of elimination but from the experience and tips that we might glean from this undertaking.

Another perhaps extreme solution is that a team that is highly experienced in machine learning and the traffic negotiation problem in general come up with their own model that prioritizes using as little as resources as possible while providing at least some degree of success in terms of performance rather than having an extremely accurate model that may take up too many resources. We listed this as unfeasible as our team as a whole probably isn’t ready to take on such a task from the ground up, and none of us are particularly that well learned especially in terms of optimizing performance and resource efficiency. There are also time constraints that prevent us from doing this in a single semester, and having a team dedicated to more time than a semester for a ground up framework is probably ideal. It is also highly likely that the price of using the time to build such a framework would be wasted, and they might be better allocated to another more relevant task that has a higher probability of success or guarantee some sort of progress.

One rather simple solution is pure brute force. Rather than a software oriented solution, one can increase the specifications of the vehicles so that rather than not having enough power we can brute force the solution. However this is probably not ideal since the Autodrive is under restraints probably, and more computing power is probably directed to other processes in the system. This is not probably going to be considered at all, as other systems are more greedy in our autodrive challenge.

4.2 Design specifications

Our team decided to optimize the current recommendation framework named HybridNets: End2End Perception Network developed by Dat Vu, Bao Ngo, and Hung Phan[10].. Computer vision for auto driving is not new in the field. Besides HybridNets, we have many other state-of-the-art networks such as: YOLO, UNet, SegNet, ERNet, LaneNet, and SCNN for lane detection and segmentation. We chose this framework, because the model claimed that they improved the speed and performance thanks to its architecture. The core concept of this architecture is to create an encoder-decoder architecture, where the backbone and neck generate a context for producing end-to-end visual perception.

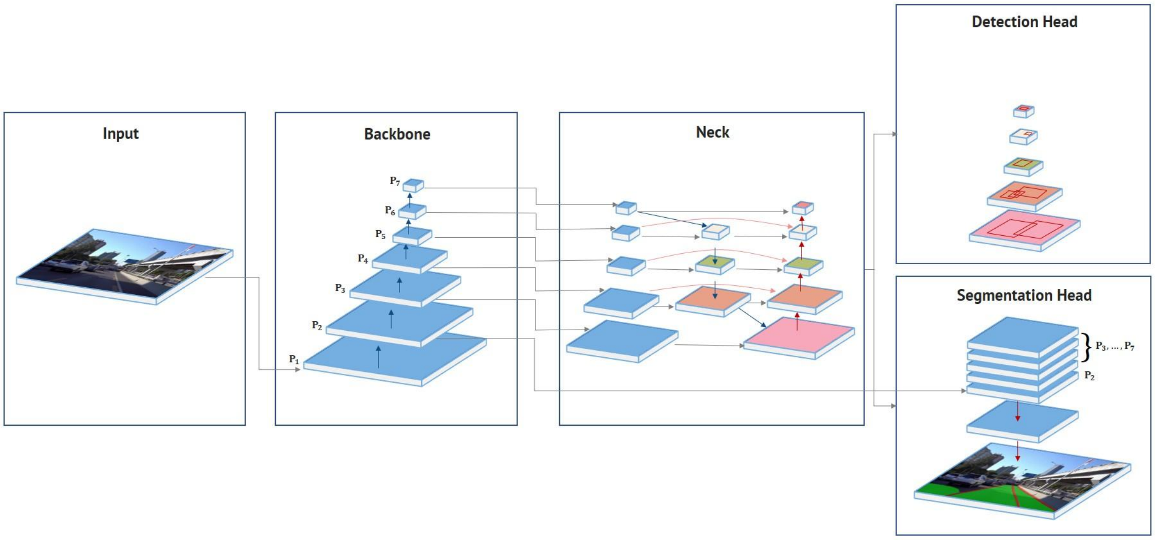


Figure 2: Machine Learning Model

General Design:

In auto driving, multi-tasking is very important. Autonomous driving must have the capacity to detect humans, weather conditions, lanes, obligations, traffic signs….Therefore, our team needs a network that can do the end-to-end multi-task visual perception. This is the core concept of HybridNets. According to the paper, the HybridNets network does traffic object detection, drivable area segmentation, and lane detection segmentation.

To achieve these tasks, the network consists of an encoder and two different decoders for multitasking. In the encoder, there are the backbone and neck. In the decoder, one is for segmentation, and the other is for detection.

Encoders:

For the whole network, the speed of the encoder is crucial because the detections and segmentation all relied on the features of the encoder.

For the backbone, we can see from the Fig. 2 that the P3 to P7 are reduced consistently in size with a resolution of $1/(2^i)$ of the input image. For example, if the input resolution is 896x384. In P2, the resolution will become $(896/2^2, 384/2^2) = (224,96)$, while in P7 it would become (7,3). They implemented these by using EfficentNet neural network models as the model proved to be efficient by its optimization in depth, width, and resolution parameters with lower computation cost. These features in the backbone will go to the neck network pipeline after extraction.

For the neck, the most challenging part is the multi-scale feature that is received from the backbone. In this paper, they use BiFPN based on EfficientDet for better performance, because it is popular in fusing features at different resolutions top-down and bottom-up and add weight to each feature to learn the importance of each level.

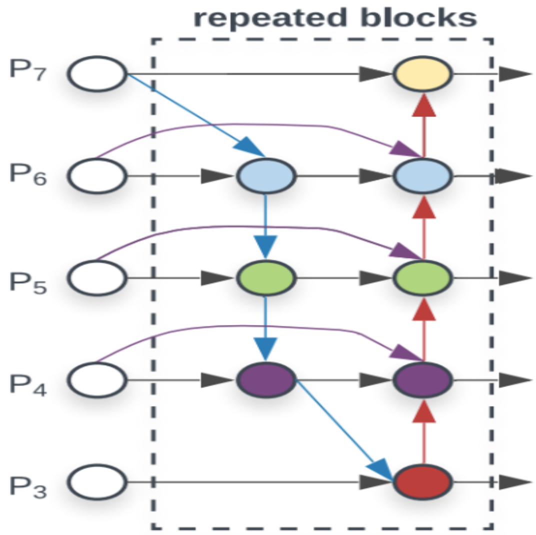


Figure 3: The BiFPN block used in neural networks.

Decoder:

As mentioned, the detection and segmentation heads are all in the decoder of the hybridNets neural network.

In this neural network, they use the idea of anchor boxes for detection. Each feature map from the neck has assigned nine prior anchor boxes with different aspect ratios. Similar to YOLOv4, they use K-means Achor clustering with nine clusters with three different scales for each grid scale in addition. In the end, the detection phase will predict the offsets of the bounding boxes from small to large. With this design, the network can smoothly on the complex dataset, and then, in the final steps, it assigns the probability of each class, and also the confidence of the prediction boxes.

In the segmentation head, they would produce three classes: background, drivable area, and lane lines. . Then, the P2 features will go through ConvBlock and meet all other features which already go through BiFPn. All Ps will be combined in the final steps to restore the final features with the original size of the image. This is where this network stands out as it helps the network improve output precision.

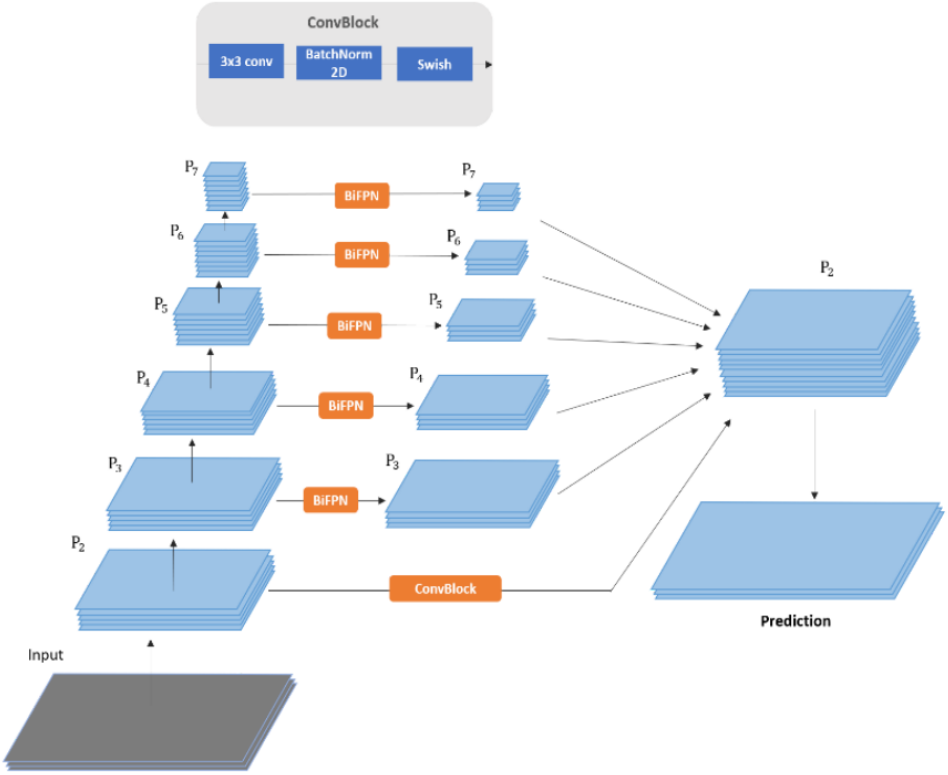
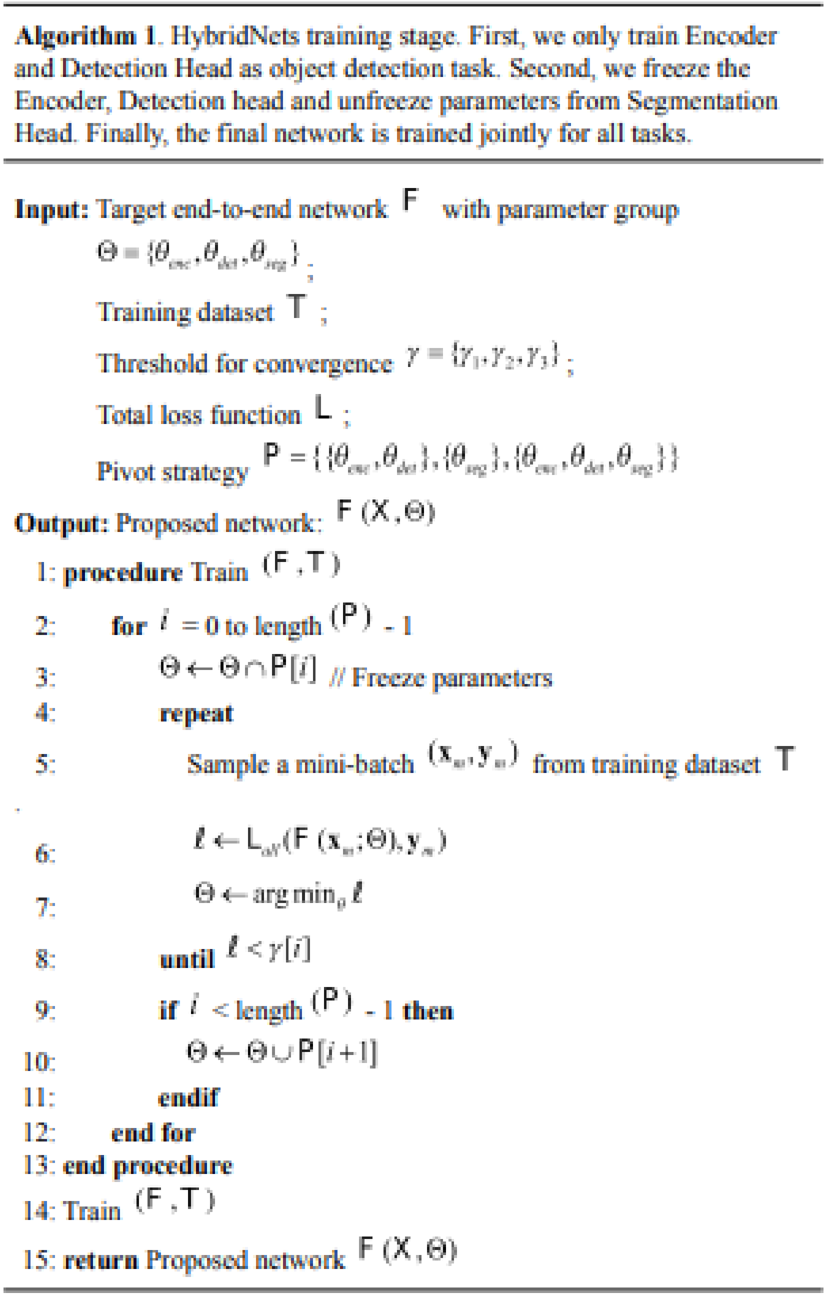


Figure 4: Neck architecture

Algorithm: The author resizes all images with resolution 640x384 before training them in the network. This resolution is a trade-off between performance and accuracy, and the dimensions are also divisible by 128 for the BiFPN. This algorithm is the logic that make the system works.



Algorithm 1. HybridNets training stage. First, we only train Encoder and Detection Head as object detection task. Second, we freeze the Encoder, Detection head and unfreeze parameters from Segmentation Head. Finally, the final network is trained jointly for all tasks.

```
Input: Target end-to-end network $F$ with parameter group
       $\Theta = \{\theta_{enc}, \theta_{det}, \theta_{seg}\}$;
       Training dataset $T$;
       Threshold for convergence $\gamma = \{\gamma_1, \gamma_2, \gamma_3\}$;
       Total loss function $L$;
       Pivot strategy $P = \{\{\theta_{enc}, \theta_{det}\}, \{\theta_{seg}\}, \{\theta_{enc}, \theta_{det}, \theta_{seg}\}\}$
Output: Proposed network: $F(X, \Theta)$
procedure Train $(F, T)$
    for $i$ = 0 to length $(P)$ - 1
        $\Theta \leftarrow \Theta \cap P[i]$ // Freeze parameters
        repeat
            Sample a mini-batch $(\mathbf{x}_m, \mathbf{y}_m)$ from training dataset $T$
            $\ell \leftarrow L_{all}(F(\mathbf{x}_m; \Theta), \mathbf{y}_m)$
            $\Theta \leftarrow \arg\min_{\theta} \ell$
        until $\ell < \gamma[i]$
        if $i$ < length $(P)$ - 1 then
            $\Theta \leftarrow \Theta \cup P[i+1]$
        endif
    end for
end procedure
Train $(F, T)$
return Proposed network $F(X, \Theta)$
```

Figure 5: HybridNets Algorithm

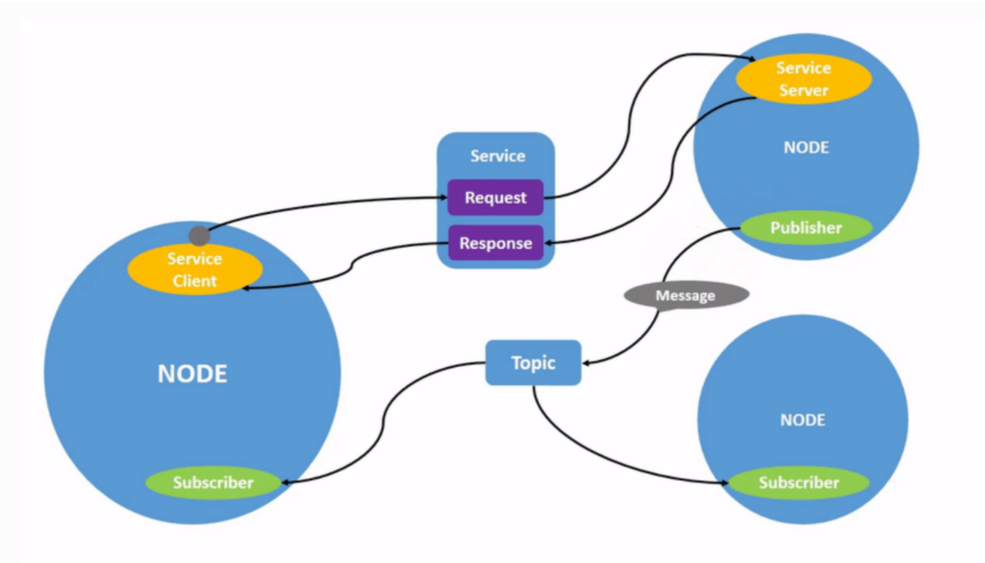


Figure 6: ROS Architecture

As shown in Figure 6, fig. taken from [8], the architecture behind ROS is a graph-like structure where nodes are processes and edges illustrate the flow of communication between nodes. In order for the team to complete the project, the team will have to create a ROS model that will better meet the needs of the project. The team formulated the below structure to better fit the scope of this project.

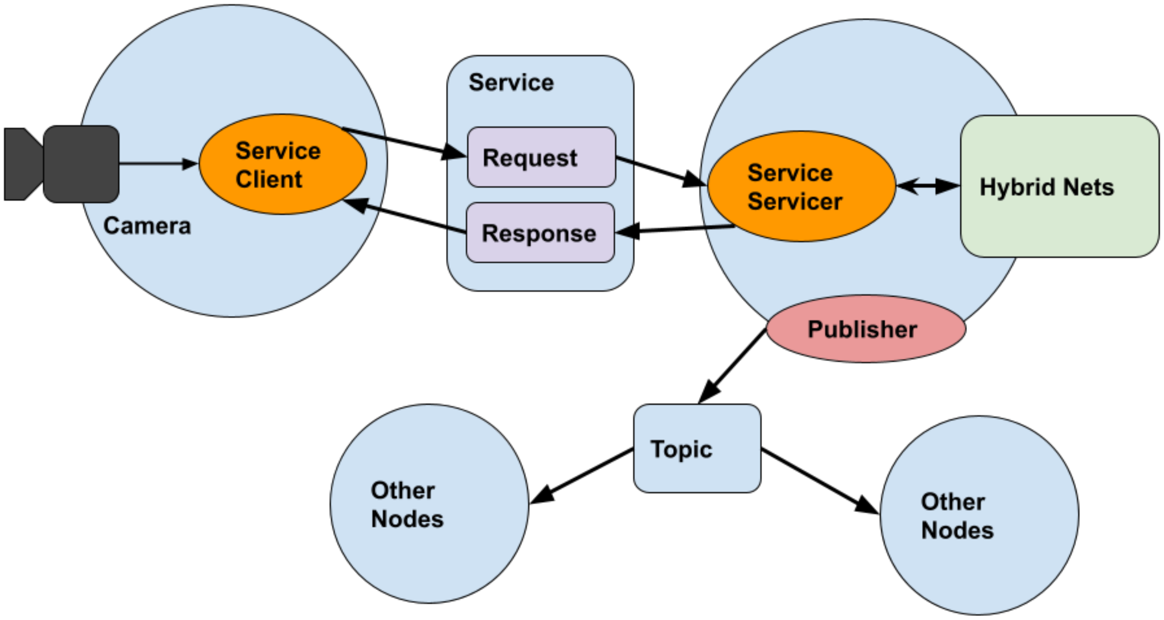


Figure 7: High-level ROS Architecture

Utilizing ROS Architecture, the team will have the machine learning model be incorporated into a ROS node. The integration of the model to the node is the main challenge of the project as the team has limited experience of the technologies at hand. The team will look into how previous models have been integrated into ROS and see if there are any libraries or tools that may aid in the integration process. Team members Jason and Khai do have experience working with Machine Learning models but ROS is a new software for all members of the team. Jason and Khai will use the foundational knowledge they learned in the classroom to better help the team in the machine learning aspect of the project. As discussed earlier in this document, the machine learning model the team will plan on integrating will be the HybridNets model due to its high accuracy and performance. The machine learning model will be placed in one node of the ROS model. Each node is meant to include only one process/purpose and thus the model being in one node is the appropriate design choice. The team will have an additional node which will capture real time information of a vehicle driving. The camera will capture video of a vehicle driving and will be the raw data that the team will use for the project. The datasets will be provided to the team so the team does not need to capture their own datasets. Figure 7 demonstrates the ROS node that is receiving input from the camera will communicate that information to the node with the Lane/Object Detection framework. The Lane/Object Framework will then utilize the Machine Learning framework to detect lanes and traffic objects. The results of the detection will then be sent to other nodes through the topic IPC method. A response will also be sent to the camera node to validate that the request was successfully processed. A topic method is used due to the fact that it is mainly used to process continuous data streams. The service method is used to be able to send a response from the node that is sending information. The other nodes are beyond the scope of this project, the main purpose of the project is to ensure that the raw input data is processed correctly. As mentioned, the exact machine learning model the team will implement will be the HybridNets model but the team will consider other models as each has their own advantages and disadvantages. A stretch goal for the team is to be able to incorporate various machine learning frameworks and do comparative testing to see which one would be best overall. However, to meet the acceptance criteria of the project, the team will integrate the HybridNets model at minimum. Regarding additional information about the nodes in ROS, it is important to note that each node has a logger. The team will utilize the logger to aid in identifying any bugs or defects that may arise from the integration process of the framework. Once the team has completed the initial integration of the model, they will then begin to do performance and comparison testing on different types of machine learning models to see which one performs best overall.

4.3 Approach for design validation

Our project has two main deliverables that we must validate to evaluate our overall success for meeting our goals. The first deliverable we must validate is making sure we have made robust and valuable comparisons between the current ROS pipeline that uses separate lane detection and object detection models and the external framework we were provided with that uses a combined lane and object detection model. One way we can make our comparisons is by simply using information and data provided by ROS and the external framework and simply compare their findings. Even though this could lead to accurate results, this is not the best approach and needs further steps to validate our results and existing data. We will make our own comparisons and validate these results by doing extensive testing using data samples on each framework. This will allow us to back-up our findings and validate our comparisons on the two frameworks on both ideal and non-ideal conditions. These tests we run will help us validate our predictions amd comparisons on accuracy and efficiency, which are the two main criteria we are trying to improve over the current ROS system. Just running the same data samples on each framework will not be enough to validate our results however. We need to run a large variety of datasets on each framework, including possibly samples included in ROS that are heavily documented, samples included in the external framework documentation, samples provided by the instructor and TA, and even samples we as a team find online or create ourselves using videos/images from streets around A&M.

The second main deliverable we must validate is that the new pipeline we built actually works as intended. This pipeline will use ROS nodes to take in images as an input, call the functions present in the external framework that uses a joint lane and object detection model, and return the new images with bounding boxes and classification results. These three steps can all be validated individually. We can validate whether the ROS nodes received the image by using ROS' existing functions to print out its nodes' input and verify it represents the image we passed in as an input. We can validate whether the external combined model framework has received the image using the same process, and also validate whether the external framework is actually reading through the image and trying to recognize lanes and objects by printing out outputs from those corresponding functions while we run the combined framework using the data sets. Finally, we can validate that the ROS nodes received the updated image with bounding boxes and classification results and that the model did perform as intended by comparing the image ROS receives at the end with the image resulting from running the external framework using the same image as input without ROS integrated. Validating all these steps will prove that we have indeed accomplished the objectives of our project and met our stated needs.

5 Engineering standards

5.1 Project management

Jason Bernal has worked at Charles Schwab as a Software Engineer Intern during Summer 2022. Jason aims to utilize the skills and collaboration he has learned through his internship and apply it to his team. Jason was selected to be the team leader for this capstone project. Jason and Khai have completed the Machine Learning course at Texas A&M University and thus are familiar with Machine Learning models. As a team lead, Jason aims to assist in all parts of the project.

Khai Nguyen has worked at Lynntech Inc. working with various technologies most notably, Computer Vision. As mentioned, Khai has an academic background in Machine Learning and will apply that knowledge to this project. Khai's credentials and experience has given him a competitive advantage of being the most familiar with the current technologies the team is tasked to work with. With such a background, Khai will be taking on the responsibility of system design. From high-level to low-level design, Khai will look into what is feasible given the time constraints imposed on the team. Khai has begun making suggestions as to how the overall architecture of the project will go.

Pavan Poladi has worked at Microsoft as a Program Management Intern. Through Pavan's work experience, he has been able to develop exceptional communication skills. Pavan is currently taking the Machine Learning class offered at Texas A&M University and will be able to apply the knowledge he gains in the classroom over the course of the semester. Pavan is assigned the responsibility of software design. Pavan enjoys solving complex problems and being incharge of software design will be a great opportunity for him to master software engineering.

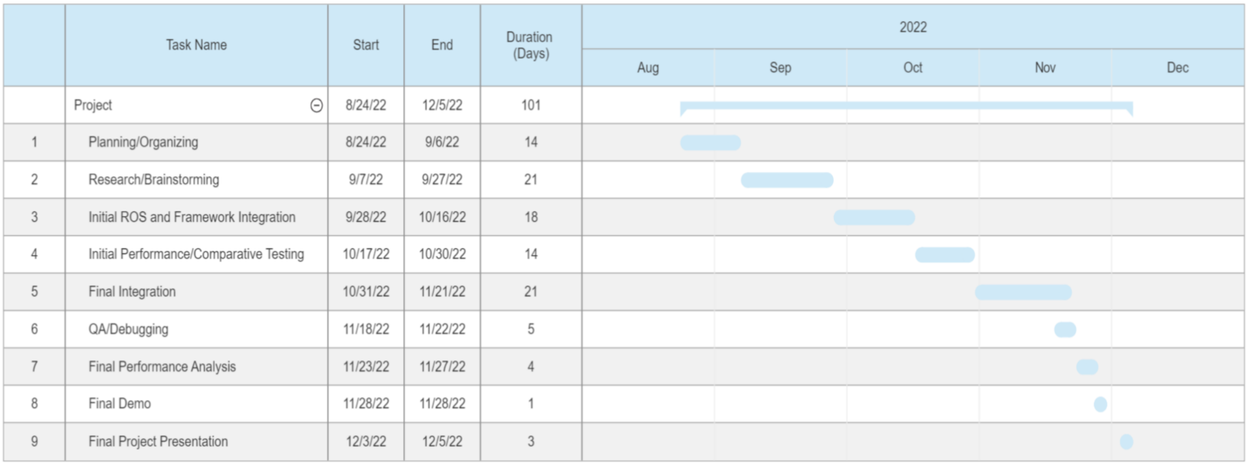
Viet Nguyen has such a strong academic background in Computer Science that he currently serves as a Peer Teacher at Texas A&M University. Viet has strong communication skills as well and is able to quickly think on his feet. Viet is assigned the responsibility of testing and validation. Viet loves analyzing systems and thus this is a great opportunity for him to work with a large-scale application. Viet will also work in other aspects of the project so that he can continue to grow as a software engineer.

Each member of the team is responsible for technical reporting. The technical reporting will be divided evenly among the team members so that each team member will get exposure to the level of reporting that a project requires. The weekly reports are rotated among the team members every week. Although the weekly report is assigned to a particular individual on the team, each member is encouraged to share the documents and discuss with the team before submitting. The team will revise and give each other feedback regarding our technical reporting to ensure we meet the acceptance criteria required.

Weekly meetings with the Instructor and TA will be vital to the success of this project. The Instructor will oversee the progress of the team and will assist if the team encounters a roadblock. In order to keep track of project progress, the team will create a list of tasks to complete and list them in the weekly report. The weekly report will specify the tasks that need to be completed and the deadline set by the team. Gantt charts will be also be utilized in order to visualize the progress of the team. The Gantt chart created will give a high level overview of the stages of progress the team will encounter over the course of the semester.

5.2 Schedule of tasks, Pert and Gantt charts

In order for the team to be able to complete the project within the allocated time, the project will be divided into different stages. The creation of a Gantt Chart will be utilized in order to better represent these stages. The Gantt chart below, Figure 8, will show the stages for this project:



	Task Name	Start	End	Duration (Days)
	Project	8/24/22	12/5/22	101
1	Planning/Organizing	8/24/22	9/6/22	14
2	Research/Brainstorming	9/7/22	9/27/22	21
3	Initial ROS and Framework Integration	9/28/22	10/16/22	18
4	Initial Performance/Comparative Testing	10/17/22	10/30/22	14
5	Final Integration	10/31/22	11/21/22	21
6	QA/Debugging	11/18/22	11/22/22	5
7	Final Performance Analysis	11/23/22	11/27/22	4
8	Final Demo	11/28/22	11/28/22	1
9	Final Project Presentation	12/3/22	12/5/22	3

Figure 8: Gantt Chart of Tasks

The beginning stages of the project involve a considerable amount of planning/organizing and researching/brainstorming ideas on how to tackle the project. ROS is a new type of software for each member of the team and thus will take time for the team to become familiar with. After the team feels more acquainted with the ROS software and Machine Learning framework, the team will move onto initial integration of the systems. It is expected that initial integration will be complex but the team will work together to meet the deadlines. When analyzing dependencies based on the Gantt chart, we can see that Initial ROS and Framework Integration must be completed first before Performance/Comparative Testing can be done. The team will need to find a way to integrate the software and then be able to measure the performance of the machine learning framework. Although comparative testing is a stretch goal for the team, the team will ensure that performance testing is done on the project. The team will report aspects such as accuracy of the machine learning framework. Another dependency to consider is the Final Integration and QA/Debugging stages. Final Integration must be complete before the team can move onto QA/Debugging. Once the team has decided on the way the integration will be implemented, the team can move onto testing the integration and ensuring that there are minimal defects. Once the team is satisfied with the correctness of the integration, the team will wrap up the project and present it to the Instructor. The final stages are reserved for completing the deliverables of the project. The Gantt Chart was an essential tool in determining the dependencies of the project.

The main critical path includes Initial ROS and Framework Integration, Initial Performance/Comparative Testing, Final Integration, QA/Debugging, Final Performance Analysis, Final Demo, and Final Presentation. The minimum time required to complete this project will take at least 65 days as suggested from our Gantt chart. The team has a total of 101 days to complete the project. The above Gantt chart depicts estimations of the time of completion for the project. The tasks on the Gantt chart have accounted for any roadblocks the team may encounter.

Regarding tasks, the team will meet every week to create a list of tasks that need to be completed for that week. Each task will be assigned to the person who best fits that role. As discussed in Section 5.1, the roles each member possesses will aid in assigning tasks. Each member is encouraged to collaborate together to collectively move forward. The team will closely monitor the tasks that are completed within the given timeframe and determine which would need to be pushed back if needed. Each member will be held accountable for their assigned tasks. The Gantt chart will be updated in terms of percentage completed when needed. The team will prioritize tasks based on the critical path of the project to ensure that proper progress is made. It is possible that requirements may change regarding the project and thus the team will remain flexible to account for any changes.

5.3 Economic analysis

Economic viability:

Cars are everywhere, and there are traffic lights and lanes everywhere, so the marketability of traffic light negotiation and lane detection are undeniably viable. Electric and hybrid vehicles (autonomous cars are mostly electric and hybrids) are becoming more popular especially with how gas prices have been increasing recently. This means that autonomous vehicles are constantly improving since electric vehicles are relatively new when compared to gas vehicles which means it has potentially a lot more room to improve. Of course, right now there is a high demand for electric vehicles but not enough supply [1]. What might be hard to market right now is that upfront costs are high due to home charging stations and initial high price point. [1] However, there are electric vehicles that may qualify for tax credit which can make the decision more enticing [2]. In addition, with the eco-friendly view that electric vehicles can present, people may be tempted to switch.

Sustainability:

System parts are available from many vendors as our project is mostly software, since we will just be only working ROS/ROS2 modules. Eventually the software may want updates in terms of new datasets as for the far future perhaps roads and traffic lights may be remodeled depending on the laws and regulations. As time goes on, accuracy and performance may need to be improved to fit requirements that laws may require autonomous vehicles to have a certain threshold for detection and negotiation.

Manufacturing:

Component tolerances might be a concern because vehicles will get better hardware as cars are constantly improving so it is important to research how different hardware components might be affected by the software. It is possible that different specifications might be used so one might need to compare how negotiation runs on different environments outside of a controlled environment. The worst case scenario is the integration of traffic negotiation and traffic light detection API doesn’t work well as having two individual modules for ROS, which means we might have to fall back on that. Compliance to regulations is very important because we are aiming for a SAE Level 4 per the Autodrive Challenge, where you aren’t driving when the autodrive is engaged, not taking over, driving under limited conditions, driverless taxis are possible, and such. From what we understand, our program will be implemented onto microchips, and for the purposes of this particular project, we don’t need it to work on all sorts of microcontrollers unless going mass production of it potentially.

5.4 Societal, safety and environmental analysis

Time is a precious commodity in today's world. The project can provide much quality of life since people spend much of their lives commuting, and the less human input involved in this can give people more free time to use in our limited lifespan. The trolley problem is a commonly discussed issue when talking about autonomous vehicles. We have to acknowledge that autonomous cars cannot have full detection of the environment due to their limitations [3]. In the context of an individual module, this project has the potential to cause mistakes or mislabel detection, for example, recognizing a sidewalk as a traffic lane, which could cost lives. As a high level, this project furthers the ambition of autonomous vehicles, which has a couple of societal implications. One such implication is that in an ideal world, people don't own individual cars but people share autonomous vehicles. This way, cars have a much higher uptime instead of staying parked at your workplace or at home. Privacy is a valid concern because cars will most likely use GPS in conjunction with image processing to have a better idea of what is coming their way for better responses. There is the question of the tradeoff of privacy for better performance and convenience. Our project does not affect pollution individually but when used as part of their intended functionality, as a part of an autonomous vehicle. According to a research done in the University of California, autonomous cars are likely to increase the vehicle miles traveled per capita through eliminating parking since cars can be running all the time, demand in car usage increasing, more optimized coordinations of cars in traffic, and allowing new travelers that couldn't afford it before [4]. On a more individual level, we can reduce waste of cars and resources by preventing crashes using traffic negotiation and objection detection.

5.5 Itemized budget

We do not foresee this project costing too much because all datasets are given for free, the fact that the project is mostly software, and that APIs are free of cost. The only potential cost that we might see is that we might choose to purchase an external hard drive to use the powerful computer we have access to in our lab room. This computer is much more computationally powerful than any of our team member's laptops, which means it will allow us to run all of the necessary models and tests we create much faster. We will outweigh the pros and cons of using this system after running some of the models and tests on our own computers first. Furthermore, we also have to consider labor hours. It is difficult to predict how long it will take to complete this project, but expecting that we spend about 10 hours of work each for the 10-week duration, this means our total labor hours needed to complete the project will be 100 hours each or 400 hours total.

Item	**Cost**
External Hard drive	50$ - 100$ depending on how big of files we need and such like datasets

6 References

[1]Horsley, Scott. “Gas Prices Got You Wanting an Electric or Hybrid Car? Well, Good Luck Finding One.” NPR, NPR, 25 Mar. 2022, www.npr.org/2022/03/25/1088287767/fuel-efficient-cars-hybrid-electric-gas-prices-surge.

[2]https://www.leafscore.com/auto/which-evs-qualify-for-the-7500-tax-credit-its-complicated/

[3] H. M. Roff, “The folly of trolleys: Ethical challenges and autonomous vehicles,” Brookings, 09-Mar-2022. [Online]. Available: https://www.brookings.edu/research/the-folly-of-trolleys-ethical-challenges-and-autonomous-vehicles/.

[4] C. J. Rodier, Travel effects and associated greenhouse gas emissions of automated vehicles. University of California, 2018.

[5] B. Liu, W. Zhao and Q. Sun, "Study of object detection based on Faster R-CNN," 2017 Chinese Automation Congress (CAC), 2017, pp. 6233-6236, doi: 10.1109/CAC.2017.8243900.

[6] M. Tan, R. Pang and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10778-10787, doi: 10.1109/CVPR42600.2020.01079.

[7] L. Haixia and L. Xizhou, "Flexible lane detection using CNNs," 2021 International Conference on Computer Technology and Media Convergence Design (CTMCD), 2021, pp. 235-238, doi: 10.1109/CTMCD53128.2021.00057.

[8] “Understanding Nodes,” *Understanding nodes - ROS 2 Documentation: Foxy documentation*. [Online]. Available: https://docs.ros.org/en/foxy/Tutorials/Beginner-CLI-Tools/Understanding-ROS2-Nodes/Understanding-ROS2-Nodes.html. [Accessed: 25-Sep-2022].

[9] *ArXiv:2208.11434v1 [CS.CV] 24 Aug 2022*. http

[10] Y. Yong and G. Xin_cheng, "A new minority kind of sample sampling method based on genetic algorithm and K-means cluster," 2012 7th International Conference on Computer Science & Education (ICCSE), 2012, pp. 126-129, doi: 10.1109/ICCSE.2012.6295041.

7 Appendices

7.1 Product datasheets

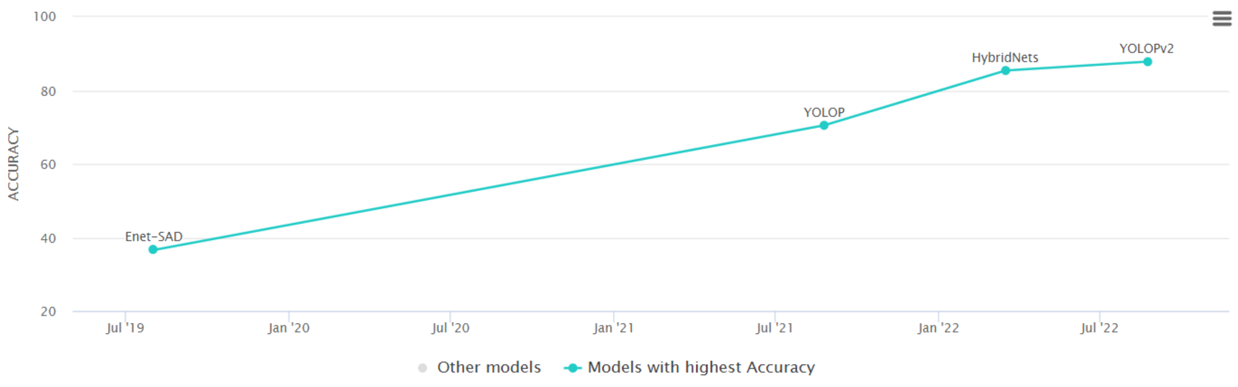


Figure 1: Comparison of Frameworks

HybridNets Paper (Proposed Model)

https://paperswithcode.com/paper/hybridnets-end-to-end-perception-network-1

YOLOPV2 Paper (Alternative Solution Model)

https://paperswithcode.com/paper/yolopv2-better-faster-stronger-for-panoptic

This one is particularly useful because it also contains comparisons to Hybridnets.

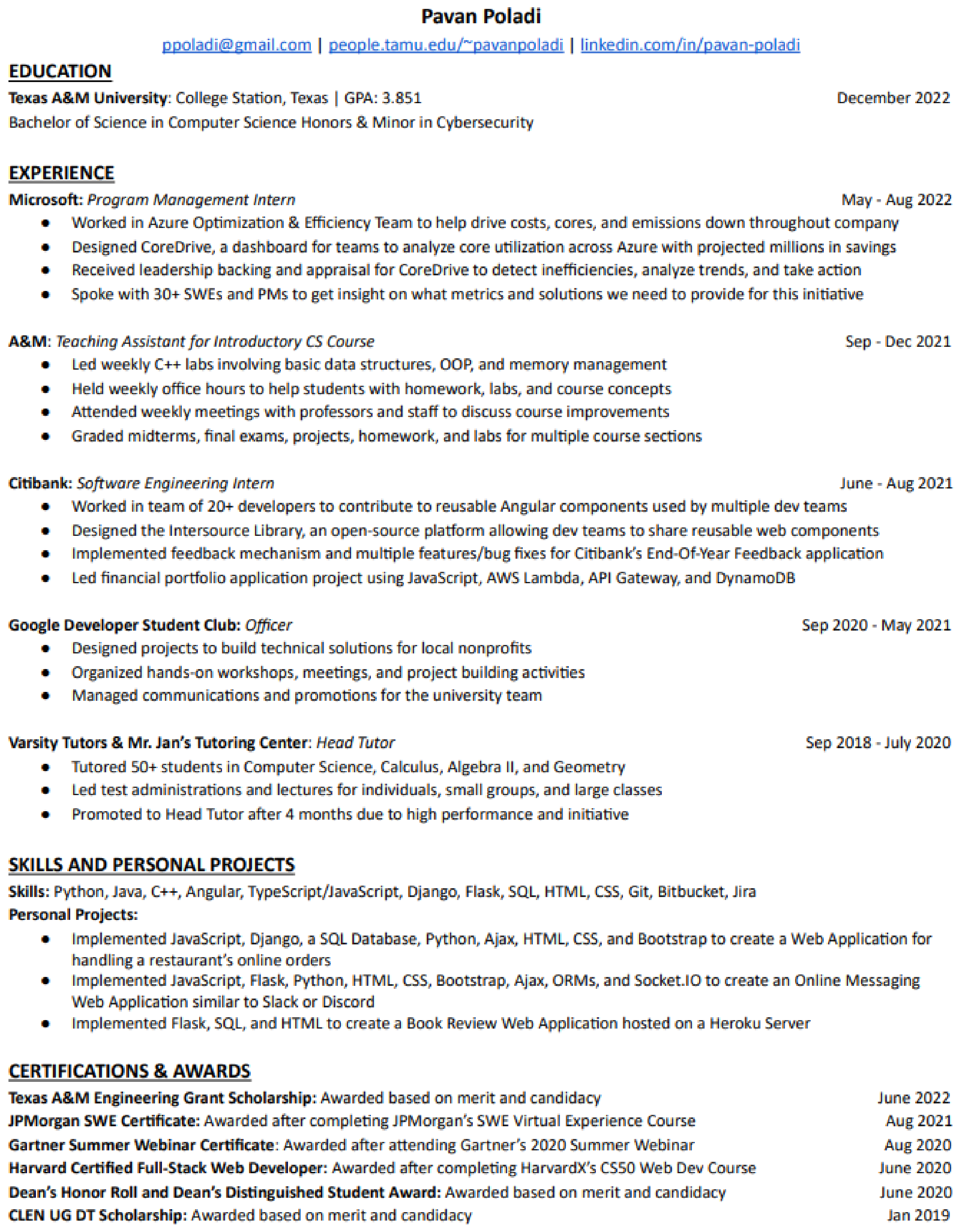
7.2 Bios and CVs

Pavan Poladi: My name is Pavan Poladi. I am a senior computer science major at Texas A&M minoring in cybersecurity. I have previously worked at Microsoft as a program management intern working in the Azure Optimization & Efficiency Team. I have also worked at Citibank as a software engineering intern creating reusable angular components for the company's online website. I also have experience as a teaching assistant for an intro CS course called CSCE 121 and making a couple of full-stack web apps in my free time.

Jason Bernal: My name is Jason Bernal and I am a Senior Computer Science student at Texas A&M University. I have previously interned at Charles Schwab as a Software Engineer where I worked under the Wire Transfers team. I worked on both frontend and backend aspects regarding wire transfers. I have also interned at ShadowSoft Inc. where I was a Python developer. My Python script automated the migration of the company's code database to Git. I am the assigned team leader for this capstone project and I look forward to working with the team.

Khai Nguyen: Innovative senior Computer Science student performing activities, including writing, and testing code, helping to develop software solutions and updating knowledge by using the latest development tools. Working in Lynntech Inc on variety projects in Computer Vision, AI, Website Management/Development, and Software Development. Capable of understanding and interpreting complex project requirements, delivering optimum customer service, working with teams to achieve common goals, and advancing organizational objectives.

Viet Nguyen: Aspiring software developer at Texas A&M University. Currently an active peer teacher for Computer Systems and hosts a weekly review for students in the course. I have been studying Japanese for nearly a year now in my free time. I hope to gain more experience and knowledge in my technical skills to increase my value in a company setting.



Pavan Poladi

ppoladi@gmail.com | people.tamu.edu/~pavanpoladi | linkedin.com/in/pavan-poladi

EDUCATION

Texas A&M University: College Station, Texas | GPA: 3.851 December 2022

Bachelor of Science in Computer Science Honors & Minor in Cybersecurity

EXPERIENCE

Microsoft: *Program Management Intern* May - Aug 2022

- Worked in Azure Optimization & Efficiency Team to help drive costs, cores, and emissions down throughout company
- Designed CoreDrive, a dashboard for teams to analyze core utilization across Azure with projected millions in savings
- Received leadership backing and appraisal for CoreDrive to detect inefficiencies, analyze trends, and take action
- Spoke with 30+ SWEs and PMs to get insight on what metrics and solutions we need to provide for this initiative

A&M: *Teaching Assistant for Introductory CS Course* Sep - Dec 2021

- Led weekly C++ labs involving basic data structures, OOP, and memory management
- Held weekly office hours to help students with homework, labs, and course concepts
- Attended weekly meetings with professors and staff to discuss course improvements
- Graded midterms, final exams, projects, homework, and labs for multiple course sections

Citibank: *Software Engineering Intern* June - Aug 2021

- Worked in team of 20+ developers to contribute to reusable Angular components used by multiple dev teams
- Designed the Intersource Library, an open-source platform allowing dev teams to share reusable web components
- Implemented feedback mechanism and multiple features/bug fixes for Citibank's End-Of-Year Feedback application
- Led financial portfolio application project using JavaScript, AWS Lambda, API Gateway, and DynamoDB

Google Developer Student Club: *Officer* Sep 2020 - May 2021

- Designed projects to build technical solutions for local nonprofits
- Organized hands-on workshops, meetings, and project building activities
- Managed communications and promotions for the university team

Varsity Tutors & Mr. Jan's Tutoring Center: *Head Tutor* Sep 2018 - July 2020

- Tutored 50+ students in Computer Science, Calculus, Algebra II, and Geometry
- Led test administrations and lectures for individuals, small groups, and large classes
- Promoted to Head Tutor after 4 months due to high performance and initiative

SKILLS AND PERSONAL PROJECTS

Skills: Python, Java, C++, Angular, TypeScript/JavaScript, Django, Flask, SQL, HTML, CSS, Git, Bitbucket, Jira

Personal Projects:

- Implemented JavaScript, Django, a SQL Database, Python, Ajax, HTML, CSS, and Bootstrap to create a Web Application for handling a restaurant's online orders
- Implemented JavaScript, Flask, Python, HTML, CSS, Bootstrap, Ajax, ORMs, and Socket.IO to create an Online Messaging Web Application similar to Slack or Discord
- Implemented Flask, SQL, and HTML to create a Book Review Web Application hosted on a Heroku Server

CERTIFICATIONS & AWARDS

Texas A&M Engineering Grant Scholarship: Awarded based on merit and candidacy June 2022

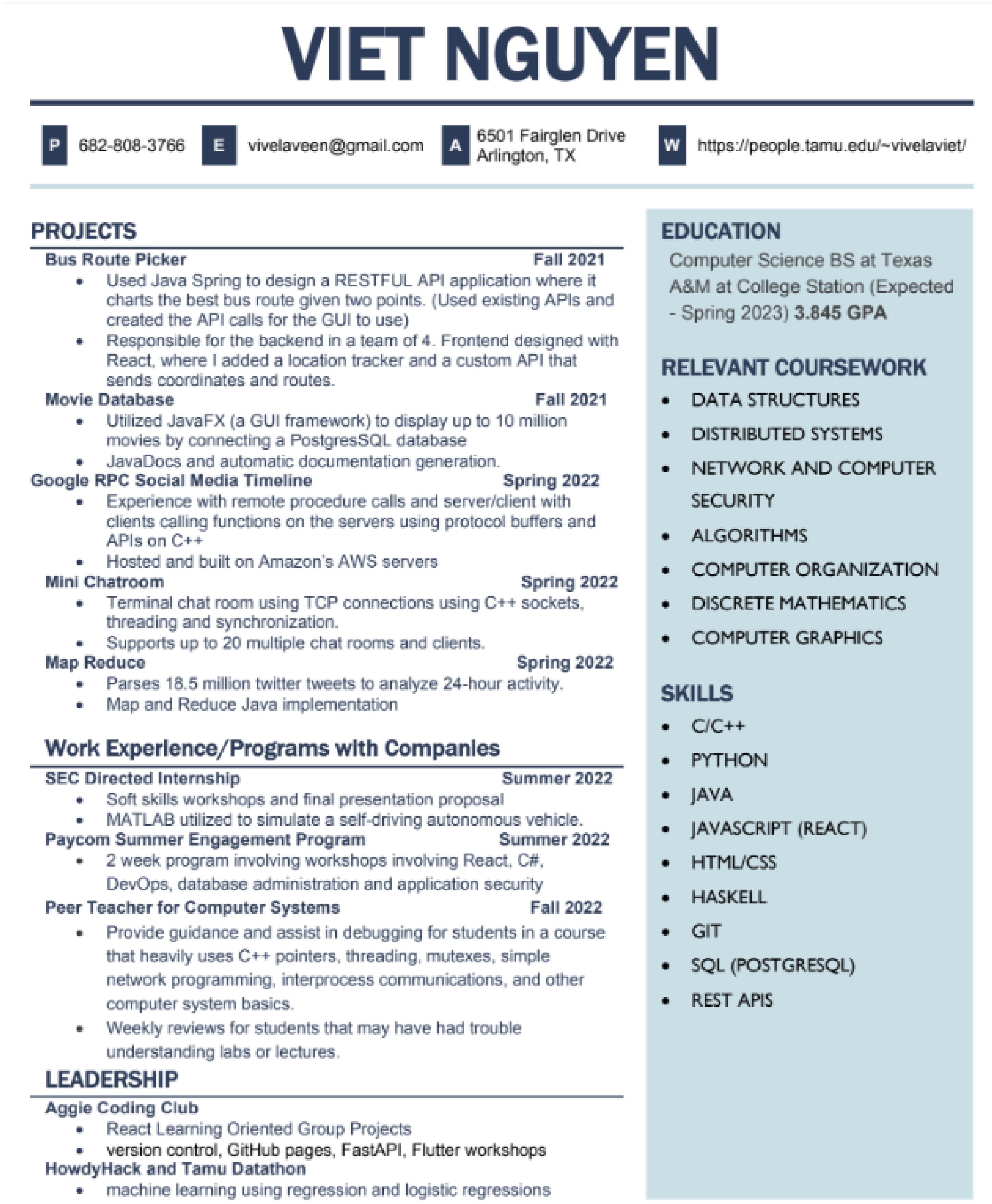
JPMorgan SWE Certificate: Awarded after completing JPMorgan's SWE Virtual Experience Course Aug 2021

Gartner Summer Webinar Certificate: Awarded after attending Gartner's 2020 Summer Webinar Aug 2020

Harvard Certified Full-Stack Web Developer: Awarded after completing HarvardX's CS50 Web Dev Course June 2020

Dean's Honor Roll and Dean's Distinguished Student Award: Awarded based on merit and candidacy June 2020

CLEN UG DT Scholarship: Awarded based on merit and candidacy Jan 2019



VIET NGUYEN

P 682-808-3766 E vivelaveen@gmail.com A 6501 Fairglen Drive
Arlington, TX W https://people.tamu.edu/~vivelaviet/

PROJECTS

Bus Route Picker **Fall 2021**

- Used Java Spring to design a RESTFUL API application where it charts the best bus route given two points. (Used existing APIs and created the API calls for the GUI to use)
- Responsible for the backend in a team of 4. Frontend designed with React, where I added a location tracker and a custom API that sends coordinates and routes.

Movie Database **Fall 2021**

- Utilized JavaFX (a GUI framework) to display up to 10 million movies by connecting a PostgresSQL database
- JavaDocs and automatic documentation generation.

Google RPC Social Media Timeline **Spring 2022**

- Experience with remote procedure calls and server/client with clients calling functions on the servers using protocol buffers and APIs on C++
- Hosted and built on Amazon's AWS servers

Mini Chatroom **Spring 2022**

- Terminal chat room using TCP connections using C++ sockets, threading and synchronization.
- Supports up to 20 multiple chat rooms and clients.

Map Reduce **Spring 2022**

- Parses 18.5 million twitter tweets to analyze 24-hour activity.
- Map and Reduce Java implementation

Work Experience/Programs with Companies

SEC Directed Internship **Summer 2022**

- Soft skills workshops and final presentation proposal
- MATLAB utilized to simulate a self-driving autonomous vehicle.

Paycom Summer Engagement Program **Summer 2022**

- 2 week program involving workshops involving React, C#, DevOps, database administration and application security

Peer Teacher for Computer Systems **Fall 2022**

- Provide guidance and assist in debugging for students in a course that heavily uses C++ pointers, threading, mutexes, simple network programming, interprocess communications, and other computer system basics.
- Weekly reviews for students that may have had trouble understanding labs or lectures.

LEADERSHIP

Aggie Coding Club

- React Learning Oriented Group Projects
- version control, GitHub pages, FastAPI, Flutter workshops

HowdyHack and Tamu Datathon

- machine learning using regression and logistic regressions

EDUCATION

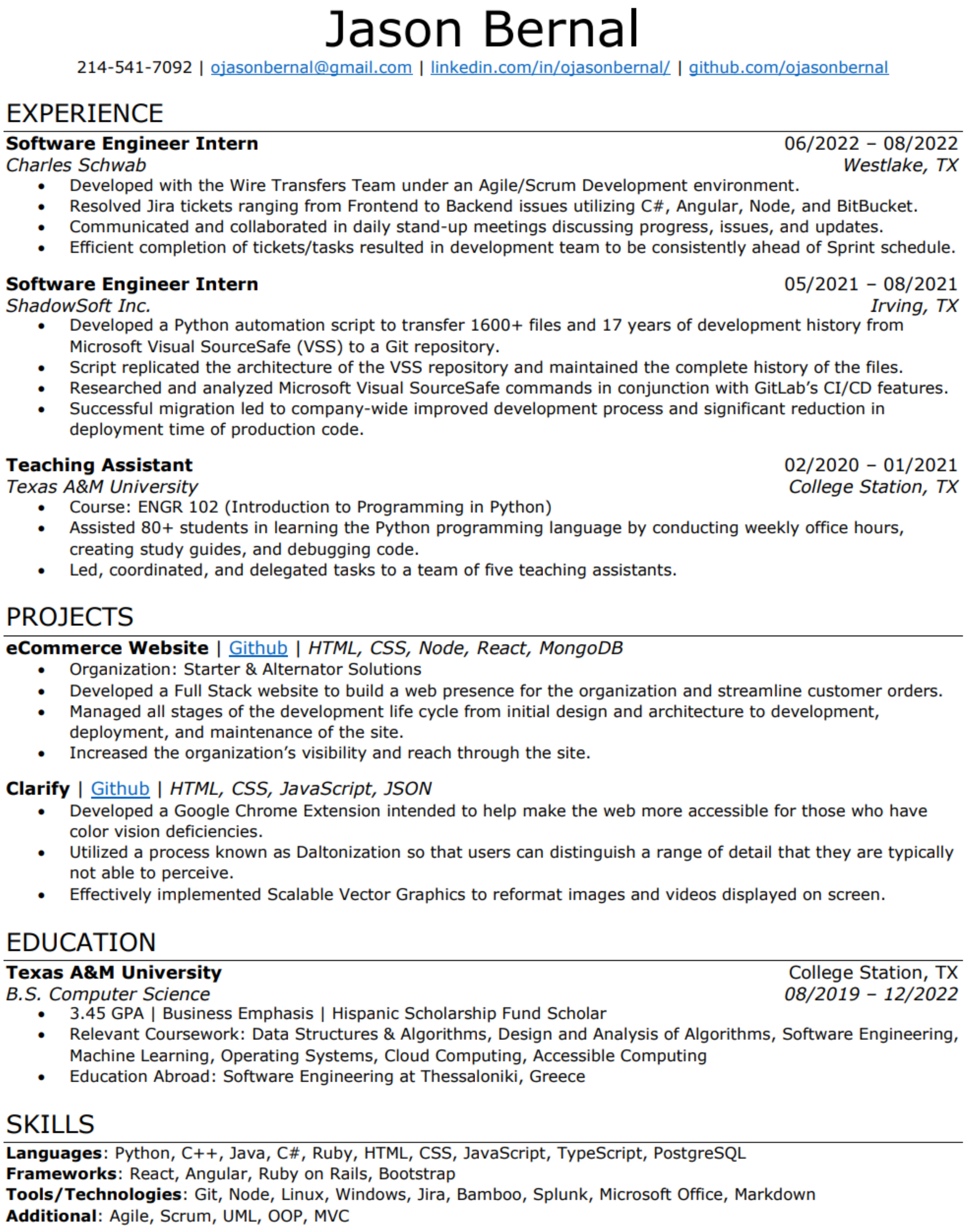
Computer Science BS at Texas A&M at College Station (Expected - Spring 2023) **3.845 GPA**

RELEVANT COURSEWORK

- DATA STRUCTURES
- DISTRIBUTED SYSTEMS
- NETWORK AND COMPUTER SECURITY
- ALGORITHMS
- COMPUTER ORGANIZATION
- DISCRETE MATHEMATICS
- COMPUTER GRAPHICS

SKILLS

- C/C++
- PYTHON
- JAVA
- JAVASCRIPT (REACT)
- HTML/CSS
- HASKELL
- GIT
- SQL (POSTGRESQL)
- REST APIS



Jason Bernal

214-541-7092 | ojasonbernal@gmail.com | linkedin.com/in/ojasonbernal/ | github.com/ojasonbernal

EXPERIENCE

Software Engineer Intern 06/2022 – 08/2022
Charles Schwab *Westlake, TX*

- Developed with the Wire Transfers Team under an Agile/Scrum Development environment.
- Resolved Jira tickets ranging from Frontend to Backend issues utilizing C#, Angular, Node, and BitBucket.
- Communicated and collaborated in daily stand-up meetings discussing progress, issues, and updates.
- Efficient completion of tickets/tasks resulted in development team to be consistently ahead of Sprint schedule.

Software Engineer Intern 05/2021 – 08/2021
ShadowSoft Inc. *Irving, TX*

- Developed a Python automation script to transfer 1600+ files and 17 years of development history from Microsoft Visual SourceSafe (VSS) to a Git repository.
- Script replicated the architecture of the VSS repository and maintained the complete history of the files.
- Researched and analyzed Microsoft Visual SourceSafe commands in conjunction with GitLab's CI/CD features.
- Successful migration led to company-wide improved development process and significant reduction in deployment time of production code.

Teaching Assistant 02/2020 – 01/2021
Texas A&M University *College Station, TX*

- Course: ENGR 102 (Introduction to Programming in Python)
- Assisted 80+ students in learning the Python programming language by conducting weekly office hours, creating study guides, and debugging code.
- Led, coordinated, and delegated tasks to a team of five teaching assistants.

PROJECTS

eCommerce Website | Github | *HTML, CSS, Node, React, MongoDB*

- Organization: Starter & Alternator Solutions
- Developed a Full Stack website to build a web presence for the organization and streamline customer orders.
- Managed all stages of the development life cycle from initial design and architecture to development, deployment, and maintenance of the site.
- Increased the organization's visibility and reach through the site.

Clarify | Github | *HTML, CSS, JavaScript, JSON*

- Developed a Google Chrome Extension intended to help make the web more accessible for those who have color vision deficiencies.
- Utilized a process known as Daltonization so that users can distinguish a range of detail that they are typically not able to perceive.
- Effectively implemented Scalable Vector Graphics to reformat images and videos displayed on screen.

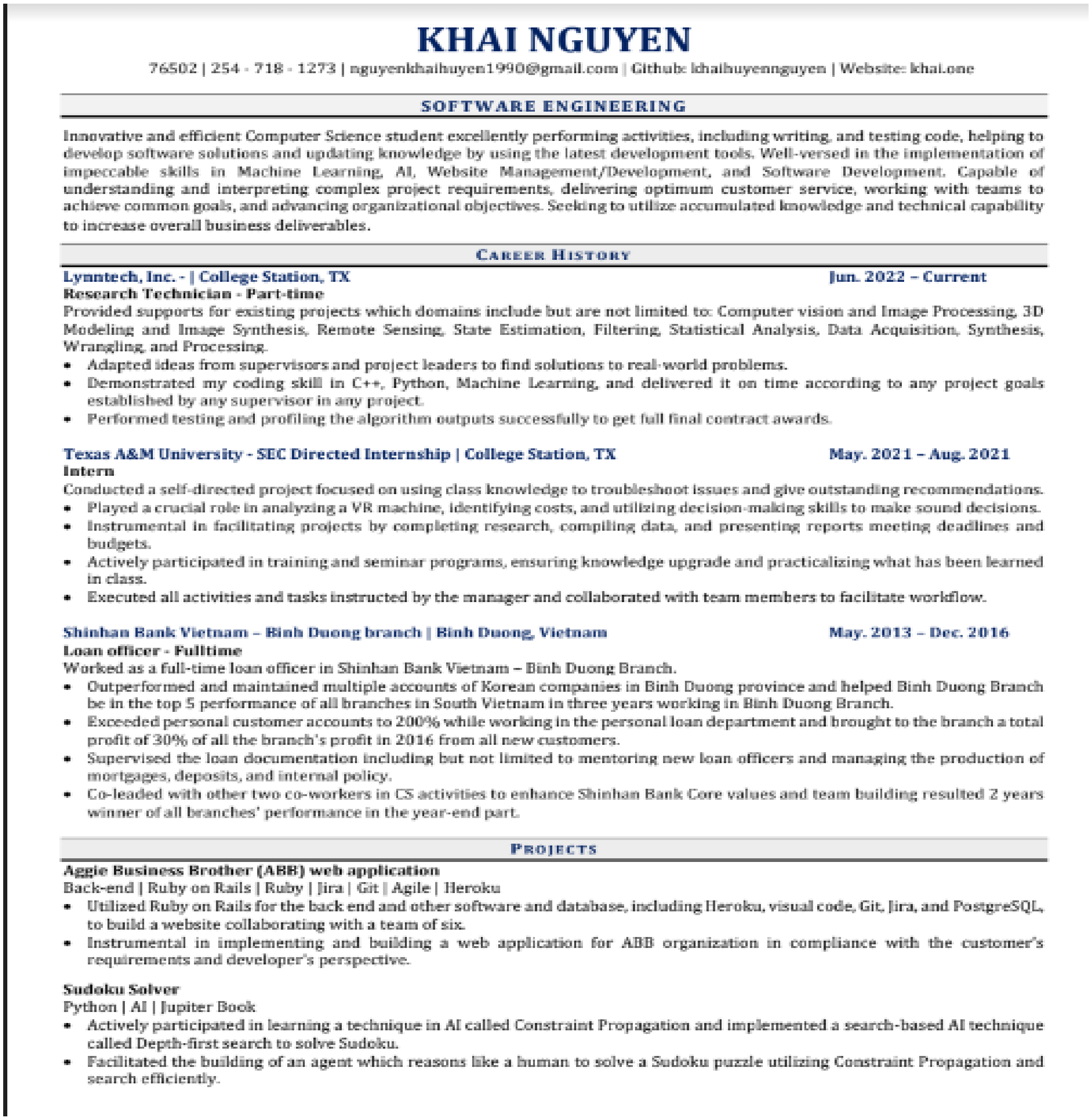
EDUCATION

Texas A&M University College Station, TX
B.S. Computer Science *08/2019 – 12/2022*

- 3.45 GPA | Business Emphasis | Hispanic Scholarship Fund Scholar
- Relevant Coursework: Data Structures & Algorithms, Design and Analysis of Algorithms, Software Engineering, Machine Learning, Operating Systems, Cloud Computing, Accessible Computing
- Education Abroad: Software Engineering at Thessaloniki, Greece

SKILLS

Languages: Python, C++, Java, C#, Ruby, HTML, CSS, JavaScript, TypeScript, PostgreSQL
Frameworks: React, Angular, Ruby on Rails, Bootstrap
Tools/Technologies: Git, Node, Linux, Windows, Jira, Bamboo, Splunk, Microsoft Office, Markdown
Additional: Agile, Scrum, UML, OOP, MVC



KHAI NGUYEN

76502 | 254 - 718 - 1273 | nguyenkhaihuyen1990@gmail.com | Github: khaihuyennguyen | Website: khai.one

SOFTWARE ENGINEERING

Innovative and efficient Computer Science student excellently performing activities, including writing, and testing code, helping to develop software solutions and updating knowledge by using the latest development tools. Well-versed in the implementation of impeccable skills in Machine Learning, AI, Website Management/Development, and Software Development. Capable of understanding and interpreting complex project requirements, delivering optimum customer service, working with teams to achieve common goals, and advancing organizational objectives. Seeking to utilize accumulated knowledge and technical capability to increase overall business deliverables.

CAREER HISTORY

Lynntech, Inc. - | College Station, TX **Jun. 2022 – Current**

Research Technician - Part-time

Provided supports for existing projects which domains include but are not limited to: Computer vision and Image Processing, 3D Modeling and Image Synthesis, Remote Sensing, State Estimation, Filtering, Statistical Analysis, Data Acquisition, Synthesis, Wrangling, and Processing.

- Adapted ideas from supervisors and project leaders to find solutions to real-world problems.
- Demonstrated my coding skill in C++, Python, Machine Learning, and delivered it on time according to any project goals established by any supervisor in any project.
- Performed testing and profiling the algorithm outputs successfully to get full final contract awards.

Texas A&M University - SEC Directed Internship | College Station, TX **May. 2021 – Aug. 2021**

Intern

Conducted a self-directed project focused on using class knowledge to troubleshoot issues and give outstanding recommendations.

- Played a crucial role in analyzing a VR machine, identifying costs, and utilizing decision-making skills to make sound decisions.
- Instrumental in facilitating projects by completing research, compiling data, and presenting reports meeting deadlines and budgets.
- Actively participated in training and seminar programs, ensuring knowledge upgrade and practicalizing what has been learned in class.
- Executed all activities and tasks instructed by the manager and collaborated with team members to facilitate workflow.

Shinhan Bank Vietnam – Binh Duong branch | Binh Duong, Vietnam **May. 2013 – Dec. 2016**

Loan officer - Fulltime

Worked as a full-time loan officer in Shinhan Bank Vietnam – Binh Duong Branch.

- Outperformed and maintained multiple accounts of Korean companies in Binh Duong province and helped Binh Duong Branch be in the top 5 performance of all branches in South Vietnam in three years working in Binh Duong Branch.
- Exceeded personal customer accounts to 200% while working in the personal loan department and brought to the branch a total profit of 30% of all the branch's profit in 2016 from all new customers.
- Supervised the loan documentation including but not limited to mentoring new loan officers and managing the production of mortgages, deposits, and internal policy.
- Co-leaded with other two co-workers in CS activities to enhance Shinhan Bank Core values and team building resulted 2 years winner of all branches' performance in the year-end part.

PROJECTS

Aggie Business Brother (ABB) web application

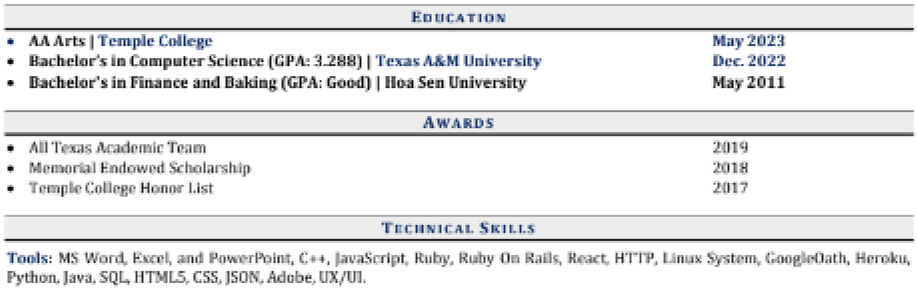
Back-end | Ruby on Rails | Ruby | Jira | Git | Agile | Heroku

- Utilized Ruby on Rails for the back end and other software and database, including Heroku, visual code, Git, Jira, and PostgreSQL, to build a website collaborating with a team of six.
- Instrumental in implementing and building a web application for ABB organization in compliance with the customer's requirements and developer's perspective.

Sudoku Solver

Python | AI | Jupiter Book

- Actively participated in learning a technique in AI called Constraint Propagation and implemented a search-based AI technique called Depth-first search to solve Sudoku.
- Facilitated the building of an agent which reasons like a human to solve a Sudoku puzzle utilizing Constraint Propagation and search efficiently.



EDUCATION

• **AA Arts | Temple College**	**May 2023**
• **Bachelor's in Computer Science (GPA: 3.288) | Texas A&M University**	**Dec. 2022**
• **Bachelor's in Finance and Baking (GPA: Good) | Hoa Sen University**	**May 2011**

AWARDS

• All Texas Academic Team	2019
• Memorial Endowed Scholarship	2018
• Temple College Honor List	2017

TECHNICAL SKILLS

Tools: MS Word, Excel, and PowerPoint, C++, JavaScript, Ruby, Ruby On Rails, React, HTTP, Linux System, GoogleOath, Heroku, Python, Java, SQL, HTML5, CSS, JSON, Adobe, UX/UI.