Allison DeCicco
Billy Miller
Lian Showl

Writing 1 Revised

Allison's Writing One

Millions of cameras are installed in houses, businesses and on street corners, recording millions of images all day. When a crime is committed, law enforcement search for the person or object they are looking for by manually reviewing hundreds of hours of video surveillance footage. However, with our product, The Average Joe Classifier, there is a way to drastically reduce the amount of time someone spends doing this task. Using this product, the officer could teach the system what he wants it to identify and upload the footage. The system will analyze the footage and label every time the specific object that it was told to find appears in the footage. This will drastically reduce the time spent doing this task and lead to solving crimes much faster. Tracking multiple objects through video is a vital issue in computer vision. It can be used in various video analysis scenarios, such as visual surveillance, sports analysis, robotic navigation, autonomous driving and medical visualization. The data that can be collected from object detection is a huge untapped potential for data analytics and could lead to advancements within many different fields.

The Average Joe Classifier will automatically identify a wide variety of objects for the user. Most cameras now track an object, but they do not identify what the object is. No product exists, as of now, that can label tracked objects. Current products for object detection have trained classifiers for specific objects. Object classifiers that currently exist are mainly for vehicles because of the abundance of traffic cameras, but it has never been expanded to have any person, whether he is a trained computer scientist or not, to classify whatever the user is interested in. The Average Joe Classifier is intended to be a generalized algorithm that will train a neural network on any item they desire. Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. The Average Joe Classifier will overcome the requirement of having a professional to have to train a neural network. The objective is to simplify a complex system, so anyone can use it. Anyone using our product will be able to identify whatever object, person or image they want. Our product will also work on a small data set, which is not common. When training neural networks, a large data set is usually required. The impact that this product could have on so many different fields is what makes it unique. It can also progress self-driving cars by helping them identify objects in their path much faster, leading to a safer solution in self-driving cars. If researchers want to track a certain animal and figure out their daily patterns, they can identify the animal and train the network, and then the system can label the animal every time it appears on the camera. Labeling images is the next big step in computer vision because it will have major improvements in various fields.

To use our product, the user will submit a video or image and tell the program what it wants to identify. The software will then run and display the results to the user. If the user is not satisfied with how the software classified items, then the user can correct it by identifying what was marked correctly or incorrectly and what was missed. Once this is done, it will retrain the neural network and then show the new results. If the user is satisfied with the results, they can save the classifier. If the user is not satisfied, it can continue to correct like previously stated until the user is satisfied. The goal is to have every successful classifier saved in a database so that users can utilize different classifiers that have already been done. This will save a lot of time for researchers or any other client that is using our product.

Finally, the product will use the YOLO algorithm. In the paper *YOLOv3: An Incremental Improvement*, the author explains the advancements in the classifier that will make this algorithm superior to other choices. Yolo uses neural networks to classify the image. It contains 54 convolutional layers. When it is trained on a set of images, it uses weights to classify the image. The YOLO algorithm solves the problem of having to use thousands of images to train a network. This algorithm is faster compared to others that are available, such as the Retina algorithm. There was a study done showing the difference and it is about four times faster than the Retina algorithm. One difficult task to tackle is when the image is not fully recognizable. The neural networks are trained on certain images, and if it is not recognizable it may be hard for the network to label the images correctly. Another challenge to overcome is creating a user interface that will make it a user-friendly experience while also transferring the information smoothly between the model and the user. The funding that we are seeking would allow the team to utilize the necessary resources to deploy the product.

Billy's Writing One

The concept of computer vision has been around since the 1960s, but in recent years the field has exploded due to advanced algorithms and powerful computers. Computer vision is now a powerful tool which is transforming industries. Computer vision is helping improve the performance of self-driving cars, allowing Google Translate to do real time translations by pointing one's phone camera at text, and helping doctors with diagnoses by analyzing x-ray images. In today's world there are cameras everywhere, from street corners to your doorbell, many of them accessible to the average person. All of these cameras pose an opportunity for computer vision to continue to grow and help more people. However, in order for an average person to harness the power of computer vision, he needs experience with programming and an understanding of computer vision. Because many people do not have this experience, our Average Joe Classifier will do the work for them.

There are many examples of how the Average Joe Classifier can help and average person. One example is if a person has a doorbell with a camera, he might want to know if the person walking up to their door is the mailman. Using the Average Joe Classifier, they would be able to train their camera to identify the mailman whenever he walks up to the door. Without the Average Joe Classifier, the user would have to write code himself to train a classifier, which the majority of people do not know how to do. The Average Joe Classifier will also be helpful with security. A casino may believe that there is a person cheating in one of the games, but in a crowded casino it can be difficult to track what that person is doing or if he is even in the casino. A casino security team probably does not have someone with computer vision skills around, but with the Average Joe Classifier it could train one of their security cameras to identify and track the person they are looking for. These are a couple instances where the Average Joe Classifier would be useful.

Not only will the Average Joe Classifier be useful, it will also be simple to use. First the user will upload a video in which the object, or objects, he wants to track appears. They will then label the object a few times so the algorithm will get an idea of what it is looking for. The model will be trained for the first time. Once it is done training, it will show a few examples of where it found object. The user will look through the examples and correct the model in the places it was incorrect. They will be able to add labels where the object was missed and remove labels that were incorrect. The model will then be retrained, and the correction process will repeat until the user is satisfied with the accuracy of his model. The user will then be able to use his model and it will also be stored in a database for future users to use if they are looking for the same object.

The process will be simple to use, but there will be several challenges in making the Average Joe Classifier a successful project. The biggest challenge for the project is the small dataset. For most computer vision projects the model is trained on hundreds or thousands of samples which improves the accuracy of the model. In our case, we know the user cannot label that many images on their own, so we will be working with a smaller dataset. Our plan for combating this problem is our iterative training approach. After the user labels a few images on their own, they will work with the Average Joe Classifier to label more images. Because the labels are generated by the system and then corrected by the user, the process will quickly create a large dataset for the model to learn from. We will also be using the You Only Look Once (YOLO) algorithm to combat the issue. The YOLO algorithm is the fastest algorithm for learning and identifying objects, so the user will not have to wait between iterations of corrections. These are just a couple of the challenges the project will face.

Once the project overcomes its challenges it will be a helpful tool. Right now, computer vision is immensely powerful and is being used for impressive applications in many fields, but it is not reaching its full potential because the technology is not accessible without the budget to hire a computer scientist. The Average Joe Classifier will make the ability to use computer vision available to everyone and allow computer vision to expand to more areas of study and for everyday uses.

Lian’s Writing One

The Average Joe Classifier: Object Detection & Recognition for Everyone

Object detection is becoming more popular in the technical world. The scope can range from personal security to productivity in the workplace. Some of the industries object detection spans is computer vision (how computers gain high-level understanding from videos/images), including image retrieval, security, surveillance, and automated vehicle systems. One of the best examples is self-driving cars which are very well-known and controversial in today’s society. Self-driving cars need object detection since it is essential in determining what the car will do next. Self-driving cars are not completely autonomous because of the judgement calls one would need to make while driving. There are still many foreign objects that the computer system in self-driving cars have not been trained on. With this in mind, self-driving cars can lead to disastrous situations without improvements on the software. For instance, when self-driving cars started in Australia, the system could not properly detect kangaroos on the road. [1] This led to having the car halt and freeze on the road, which could have ended poorly for the passenger(s) in the vehicle. Most object classifier systems use the ground as a reference point, but since kangaroos jump into the air, the system has a hard time tracking how close the kangaroos are and predicting where the landing point will be.[2] The problems in this system are urgent and must be addressed before continuing to move forward in this instance since self-driving cars are heavily pushed for daily consumer use. Without the continued research and improvements of object detection and classification, people’s lives could be at risk for not having the most recent object classification software. By investing in our team’s Average Joe Classifier, the extra improvements and research will better object recognition and detection technology for the future. Our team’s project will train and improve upon the object classifier infrastructure that has already pioneered the new age of real-time image processing. The people who interact with this technology every day would benefit from improvements to the object classifier. The technology will be accessible for more people in various real-world applications, like self-driving vehicles.

Furthermore, our project will use YOLO, which is an algorithm that is used to mean, “you only look once” which is important for fast and real-time multi-object detection. YOLO applies a neural network to the entire image and divides the image up and creates bounding boxes to be drawn around images. From there, the computer predicts what the object is from a trained database for data storage.[3] The YOLO algorithm provides most of the technical challenges that this project will face. Some of the challenges include, learning how to implement YOLO and how the algorithm works within the project's context, learning how to setup data in real-time, creating bounding boxes setup for images (bounding boxes are drawn around images by the user to help the computer classify images), and determining whether or not YOLO's results and predictions are correct or incorrect. For the beginning portion of our project, some of the technical challenges can be resolved once extraneous research is done. Our team will understand the literature that already exists for object classifiers. To find solutions for other technical challenges, our team has hypothesized that we must focus first on how to implement YOLO, how others have used it, how to modify our project to solve the issues of the existing software for object classifiers, and how to lean on linear algebra and other applied mathematics to draw bounding boxes with our software.

Overall, our Average Joe Classifier will be able to be used by the general public rather than limited to private companies that wish to use an object classifier. The classifier would be able to train whatever the user wanted to detect, so the data that the user supplies to the classifier would in turn take the data to then classify and then store the data that it has been fed. This is useful to refer back to models that have already been stored. Sometimes a user needs to access a particular classifier, for instance, a person of interest, and our classifier would in turn save the user trained input to a "penguin" classifier (classifier that detects penguins) in our database. If another user wanted to train the penguin classifier with more pictures in identifying penguins, that data would need to be stored in the respective penguin classifier folder in the database. To conclude, this project would benefit existing fields that use object classifiers and the technology that surrounds object classifiers. Improved object classifiers would fully provide the crucial and necessary classifiers to ensure the safety and accuracy to those who use it. The technological advances from the object classifier improvements will develop and impact many other disciplines that depend on its uses such as, facial recognition and detection technology, self-driving cars, object tracking, activity recognition in law enforcement and medical imaging to shape the future of these technologies.

REFERENCES:

[1] A. Luehrs, "Seven Problems Self-Driving Cars Need to Overcome", *Smith's Lawyers*, [Online document] August 2018, https://www.smithslawyers.com.au/post/self-driving-car-problems. [Accessed September 30, 2019]

[2] Ibid.

[3] J. Redmon and A Farhadi, "YOLO v3:An Incremental Improvement", *University of Washington*[Online document], 2018, https://pjreddie.com/media/files/papers/YOLOv3.pdf. [Accessed September 30, 2019].

Writing 2 Revised

The Average Joe Classifier: Object Detection and Recognition for Everyone

Similar Work

One of the most challenging issues in the field of computer vision is the issue of how to create a good model with a small dataset. One approach to solving the issue is documented in Chuck Rosenberg, Martial Hebert, and Henry Schneiderman's paper titled "Semi-Supervised Self-Training of Object Detection Models." The researchers developed a process which they call "semi-supervised learning." They start with a small set of labeled images and a larger set of images which they refer to as weakly labeled images. They begin by training their model on the small set of labeled images. During this process, they also generate statistics about the observed objects to be used with the weakly labeled set. They use the statistics to make predictions about where the object they are detecting appears in each "weakly labeled" image. Once the predictions are made, they use what they refer to as a selection metric to decide which of the predictions on the "weakly labeled" images are best. The selection metric is a way to estimate the probability that the label given to the image is correct. The top few predictions, those which scored highest on the selection metric, are then placed into the set of images that are considered fully labeled. The process then starts again, now with a larger labeled set and a smaller weakly labeled set. The process will continue to iterate until the weakly labeled image set is gone and the model is done training. They tested their results by comparing them against the results when training on a traditional fully labeled dataset. The researchers found that, in order to maintain the same accuracy as a fully labeled dataset, as the number of labeled images got smaller, the number of weakly images required got larger. However, the total number of images, labeled and weakly labeled, remained constant and similar to the number of labeled images required for a fully labeled dataset. For the researchers this is a successful result because weakly labeled images are easier to obtain than fully labeled images. The Average Joe Classifier uses a similar approach, using the model to label more images to add to the labeled set. The difference is, instead of using complicated math to determine how well the model did, the Average Joe Classifier has the user review the results and help fix the classifier's mistakes. This will be more accurate than the complex selection metric and produce the same result [1].

An example of another group of researchers approach to solving the problem that computer vision needs a large labeled dataset can be found in Ervin Teng, Rui Huang, and Bom Iannucci's paper titled "ClickBAIT-v2: Training an Object Detector in Real-Time." The goal of their research was to create a way to train a model from a live video stream on an unmanned aerial system. They are not directly solving the large dataset issue, but the amount of data is inherently small when the data is a live video. Their solution is also relevant to the Average Joe Classifier because they have the user help with labeling. Their process begins with a user clicking on the object he wants to classify in the video. They used other computer vision techniques to create a box around that object, then trained the model and obtained a result. Once they obtained their starting model, they used the assumption that the object would not have moved much between frames of a video, so they were able to create a box around the object again and continue to track it. If the model did not find the object in the next frame, the user would select it again. This process continues until the model is satisfactorily accurate. A significant difference between the ClickBAIT-v2 project and the Average Joe Classifier is that the Average Joe Classifier is more general. Because ClickBAIT-v2 was specifically designed for unmanned aerial systems, the researchers knew they would mostly be looking for people. This means that they could begin with a classifier already trained to detect humans requiring the process to take fewer iterations. The Average Joe Classifier is designed to be able to build a classifier for anything, so it does not start with any sort of knowledge beforehand. Overall, ClickBAIT-v2 and the Average Joe Classifier use similar iterative processes, but ClickBAIT-v2 is more specialized and does not rely as much on the user to help it make accurate predictions [2].

One final example of a product similar to the Average Joe Classifier is described in Nathaniel Roman and Robert Pless' paper titled "A System for Rapid Interactive Training of Object Detectors." In their work they created a product which uses a process similar to the Average Joe Classifier. They also use the process of having the user correct labels between iterations of training to create a classifier. However, when they created the product in 2009, computer vision technology was not as impressive as it is today. One of their problems was making the training and labeling process quick enough for the user to go through the process. Roman and Pless had to focus more of their energy on making the labeling process faster to compensate for the slow training process. With modern computer vision technology, the Average Joe Classifier will be able to train and label objects in real time, drastically shortening the time for the whole process and making complex technology more accessible to a wider audience [3].

Target Audience

With this in mind, the target audience for the Average Joe Classifier will impact many companies and people in different fields and/or situations. People who are interested in using our project will typically want to classify an object, animal, person, etc. Some applications are law enforcement, biology research, self-driving cars, facial recognition, medical imaging to name a few. Our target audience is broadening to average people whom may need our classifier for everyday things. These are only a few examples since our Average Joe Classifier can be used in so many other ways. The user can be creative in how our product will be used. There will be different set prices depending on the client. Our team will ensure to cater to the client's needs since some will only need the software for a few weeks or months, whereas, other clients will need the software for years. The Average Joe Classifier will have a free trial period where any user that signs up for an account can use a limited version of the software for one week only. This will be to keep some of the premium features, exclusive to our paying clients. The free trial version will allow a user to have limited usage of the classifier. For example, the user will only be limited to saving two classifiers to the database, and the client will not be able to share publicly the classifiers that have been created.

Additionally, the premium version will allow a client to have many classifiers stored in our database. The client would be allowed to share publicly with the other users of The Average Joe Classifier. Anyone is eligible to be a premium member with the monthly price of $20 per month, $100 for 6 months and $200 for the whole year. It makes sense to have the pricing incremented in this way since most clients might only need the software for projects that do not take up a lot of time. On the contrary, other clients might be larger companies and corporations that will use the software all year round. The work that the software is doing for the larger clients is monumental since our team is marketing our product at such a low price.

Furthermore, as mentioned before, law enforcement will be greatly impacted by our Average Joe Classifier. Law enforcement will be able to find items and/or people in security cameras. The Average Joe Classifier will allow law officials to track down criminals in shoplifting incidents, missing person reports, track person of interests, etc. The Average Joe Classifier can help to identify a particular person's face that can help with investigations and assist investigators to figure out criminal acts. An example of this is if a store owner's most valuable item is stolen. Instead of having the store owner looking through hours and hours of security footage with multiple cameras throughout the city, our Average Joe Classifier will be trained to find that missing item. The classifier can be set to look out for the item. Then the classifier will notify the officer or investigator to the particular camera with the classifier that believes the missing item is shown. This significantly reduces the search time for the item and increases efficiency to hopefully getting the missing item back to the store owner. This is one of the example clients that the Average Joe Classifier can be an essential part.

Similarly, the Average Joe Classifier will be beneficial to biologists in biology research. A recent study explains the use of an animal scanner called 'camera traps' and how those regular cameras malfunction when trying to study specific animals. Another issue that the biologists encountered with the software was that the cameras would take an enormous image collection and detecting the animals in the pictures were daunting [4]. Our Average Joe Classifier is a better solution over regular cameras for these biologists. One of the concerns the biologists had were with humans and other foreign objects/not the animals of interest interfere with the footage that the camera traps capture. Another challenge the biologists faced was the camera battery draining after turning on to record the human's actions (or another animal) instead of the animal of interest. This is one of the important ways that the classifier could be used in biological research.

Moreover, a person does not have to be using our Average Joe Classifier for work or research-related purposes. Average people can use this software just like big companies. This levels the playing field between two audiences (average people and larger corporations/companies) allowing for the same software to be accessible for all. For instance, some people who are retired have bird feeders in yards to watch the different birds that visit the feeder. The people who enjoy bird watching may not want to be outside all day watching since sometimes birds will be absent. In this case, a bird watcher may want to find the frequency of birds the bird feeder, but does not have enough time in the day to do that. That person might also want to set up an experiment to find which bird food is preferred. Instead of watching the bird feeder all the times of the day, the bird watcher can simply choose our Average Joe Classifier to do the classifying work. We will create awareness for our classifier in small advertisements marketed towards the 'average joe' since bigger corporations will know about our classifier through licensing.

Societal and Global Impact

Data are collected at a rapid pace, making it difficult to keep up with the analysis of these data. A substantial amount of data are collected in pictures or videos, requiring the researchers to do a time-intensive review to find what they were searching for. With the Average Joe Classifier, these data can be analyzed quickly and efficiently by someone who is not trained in computer science or machine learning. This will lead to advancements in many different fields because research is performed faster with a machine analyzing the data then having a person have to watch hours of film or scroll through and label thousands of images. The broader societal need that this product addresses is bringing a technology that is usually for someone who is trained in computer science and adapting the technology, so the average joe can use machine learning. Although machine learning can be very complicated, it is a technology anyone can adopt it. Our system is meeting the needs of people wanting to have a complicated technology. Our product could be used internationally. There is no cultural or language barrier when using the Average Joe Classifier. The user interface will allow the customer to choose the language he is most comfortable, so that there is no confusion while using the interface.

In addition, some advancements that were made with machine learning area described here. Machine learning influencing different areas of life. The following examples shown the broad range of fields machine learning is utilized. One field that is adapting machine learning to make their job more efficient is biology researchers. For example, a researcher might put a camera on a bird's nest to watch the bird's habits of when he leaves the nest to get food. Without machine learning, the researcher would have to watch hundreds of hours of film and label which time the bird left and returned. With our system, the labeling would be done for the researcher and save him the hours of having to sit at a desk and watch the footage himself. Another example where object classifiers impacted society is in video surveillance for law enforcement. In the paper *Shape Based Object Classification for Automated Video Surveillance with Feature Selection,* the authors discuss the advantages to have a highly trained classifier in surveillance. Law enforcement spends many hours searching through film for one person or object, but this can take hours. The paper states that there is a significant reduction in time for the user when he uses the object classifier. These examples all decrease the time needed to do their respective task, allowing for advancements and discoveries to be made faster.

Next, the Average Joe Classifier will not need to be regulated. However, the system will be public and since the classifiers will be stored in a database, a cyber security element will be added. This will ensure that the information that is classified is secure and someone cannot access it who is not utilizing the system properly. Our product cannot do harm directly. However, a customer could use the product with bad intent by trying to identify something in an image or video that could aid them in doing harm. A way to prevent this would be monitoring who uses the interface. To prevent users with bad intentions from using our site, we plan on verifying the customers that want to operate our product. If the user inputs their name, the name can be cross referenced from a database and if there is nothing to be concerned about after this check, then the user can be allowed to access the system. Another way to prevent harm is to verify what the customer is labeling. For example, if the user submits pictures and he is labeling something as an automatic weapon or a bomb this would be flagged, and this user would no longer be able to use our product. The classifier that the customer created would not be put into the database. These are some ways we minimize and prevent harm caused by the use of the product.

In conclusion, the Average Joe Classifier will have the greatest impact on someone who is not savvy with technology. The product will take a task that is perceived as very complicated and make the technology accessible and understandable to anyone. Since our product is so adaptable, many fields such as biology, marketing and technology can all be impacted.

References:

[1] C. Rosenberg, M. Hebert, and H. Schneiderman, “Semi-Supervised Self-Training of Object Detection Models,” *Workshops on Applications of Computer Vision*, vol. 1, 2005. [Online Serial]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4129456&tag=1/. [Accessed Oct. 1, 2019]

[2] E. Teng, R. Huangt, and B. Iannucci, “ClickBAIT-v2: Training an Object Detector in Real-Time,” *Workshops on Applications of Computer Vision*, vol. 3, 2007. [Online Serial]. Available: https://arxiv.org/pdf/1803.10358.pdf/. [Accessed Oct. 1, 2019]

[3] N. Roman and R. Pless, A System for Rapid Interactive Training of Object Detectors,” *International Symposium on Visual Computing*, 2009. [Online Serial]. Available: http://www2.seas.gwu.edu/~pless/papers/romanPlessISVC.pdf/ . [Accessed Oct. 1, 2019]

[4] H. Yousif, J. Yuan, R. Kays and Z. He, “Animal Scanner: Software for classifying humans, animals and empty frames in camera trap images,” *Wiley Ecology and Evolution*, vol. 9, Ocotber 2018. [Online serial]. Available:https://onlinelibrary.wiley.com/doi/full/10.1002/ece3.4747. [Accessed September 30, 2019].

[5] R. Hota, V. Venkoparao, “Shape Based Object Classification for Automated Video Surveillance with Feature Selection,” *10th International Conference on Information Technology (ICIT 2007),* December, 2007.[Online serial]. Available: https://ieeexplore.ieee.org/abstract/document/4418276.[Accessed September 27, 2019]

Writing 3

The Average Joe Classifier: Object Detection and Recognition for Everyone

Technical Innovation

The goal of the Average Joe Classifier is to bring the ability to leverage computer vision to the normal person. We plan on achieving this goal by creating an easy to use interface which shields the user from the complex computations involved and by designing a training process which helps the user to quickly label data instead of spending several days labeling all of the data themselves. The user interface will be built with Hyper Text Markup Language (HTML) and JavaScript, which is very common and not the innovative part of the product. We will also be using a MongoDB database to store and organize the data which is collected and generated, but this is a standard use of the technology. What makes the Average Joe Classifier innovative is the training process.

Training a model is the most difficult and time-consuming part of a computer vision project and the Average Joe Classifier aims to solve that problem. The Average Joe Classifier uses a process of training and correcting as a way of generating data faster. The idea is the classifier can be trained on only a few images and then the user can correct its mistakes on those test images. This training and correcting process will keep repeating, and with every iteration the number of mistakes should go down and the user will spend less time correcting. Soon, the user will have a full dataset of labels with minimal work. This is much more effective than the user labeling hundreds or thousands of images one by one, which a normal person does not have time to do. We believe this unique process will allow for easier access to computer vision and its applications.

Additionally, there has been research done on products which use similar iterative training approaches. Ervin Teng, Rui Huang, and Bom Iannucci wrote the paper titled "ClickBAIT-v2: Training an Object Detector in Real-Time". The goal of their research was to create a way to train a model from a live video stream on an unmanned aerial system. This is similar to our product because ClickBAIT-v2 did not have a large dataset and was depending on the interaction with the user to help label objects. This product differs from the Average Joe Classifier because ClickBAIT-v2 was a very specific classifier the authors tried to create. Our classifier will classify any image the user wants, making it more general than this product. Robert Pless also created a product similar to what we are creating. He wrote "A System for Rapid Interactive Training of Object Detectors." This paper explains how the authors went about solving the same problem the Average Joe Classifier is solving but in 2009. The technology in 2009 was not as impressive as current technology, so Pless was not able to create a finished product. Pless was also trying to label an object or person based off what the user inputted.

Although these products use similar iterative training techniques, we will rely more on the user and make use of better technology to create the Average Joe Classifier. The iterative training approach will allow any user with any skill level to generate their own computer vision model in a fraction of the time normally required.

Technical Feasibility

We believe this product is technically feasible because the core technologies building the systems are commonly used and easy to utilize and customize. HTML and JavaScript are the technologies utilized for the user interface. To organize the items displayed on the user interface, we used HTML. The library divides the page into rows and columns to organize the page. HTML gives the page structure, similar to how walls give a room structure and legs give a chair structure. Using CSS (Cascading Style Sheet) gives the page color and design. CSS also gives the page style. For example, it tells the webpage how big to make the font, and what color the buttons should be. In other words, the CSS gives the page an aesthetic. JavaScript allows for the user to interact with the webpage. It tells the webpage what the webpage's proper reaction to a click of a button should be, for example. The JavaScript simply gives the webpage functionality. Without CSS, the page would be just a screen with text.

In addition to the technologies used for the user interface, we used MongoDB for our database. A database is a tool that allows the application to store information in one place. For example, a page like Facebook will store its users' usernames and passwords in a database. This makes the information easy to access. MongoDB is a non-relational database, which means the data are not stored in a table with rows and columns, like more traditional databases. The full image or the path to where the image is located can be stored in the database. MongoDB also can easily expand its size based on the number of images that a user submits. Other databases do not expand as easily depending on what they are storing, which would create a problem for us since we are not aware at the beginning how many pictures users are going to submit for their classifier at the beginning of the project.

Next, to perform the machine learning part of our project, we used the YOLO algorithm. The YOLO (you only look once) algorithm is the driving piece for our classifier. Machine learning is an application of artificial intelligence that provides the systems the ability to automatically learn and improve experience without being explicitly programmed. This algorithm was not researched as extensively as others and therefore is not as efficient as others when Pless tried to create his product. YOLO is what is driving our product. Without YOLO, the customer would submit images, but the images would come back from the database with no label. Without a label, there is no classifier, and therefore no Average Joe Classifier. This is the most novel part of our product. Using this technology, combined with the user interface, our product will stand out amongst the other products on the market. The classifier will be given a number of images to label an object or person. The images will be put through the classifier. To run the algorithm, Python, a scripting language will do the work. Python is what interacts with the computer to put the labels on the various images.

Furthermore, the YOLO algorithm is the key feature of the Average Joe Classifier that makes this project feasible. The goal of our project is to leverage the user to create a dataset faster and more efficiently. However, we will not achieve this goal if the user has to wait for hours during each iteration of training. The YOLO algorithm solves this problem. YOLO is currently the fastest computer vision available, which is why we have chosen it for the Average Joe Classifier. We are using the YOLOv3-tiny version of the YOLO algorithm which, when compared to other algorithms, is by far the fastest. YOLOv3-tiny is able to process 244 images per second while most other algorithms do not even reach 20 images per second. YOLOv3-tiny's accuracy suffers a little due to the increased speed, so we considered other versions of the YOLO algorithm. Other versions can process around 50 images per second with better accuracy, but we decided to use YOLOv3-tiny because speed is the most important part of the Average Joe Classifier. Using YOLOv3-tiny as the algorithm for the Average Joe Classifier will allow the user to only have to wait a few minutes between iterations of training instead of hours.

In conclusion, the Average Joe Classifier utilizes HTML, CSS, and JavaScript for the user interface. To store images needed for the classifier, we used a MongoDB database and the YOLO algorithm to create labels. The YOLO component of our project will make this product novel because any user will be able to use machine learning to create a classifier. Without it, a normal person would require a large time commitment and thousands of images to create their classifier. YOLO allows our product to work around these limitations usually associated with machine learning. To run the algorithm, the scripting language Python does the task. These products work together to create the Average Joe Classifier.

Cost, Risks, and Risk Mitigation

There will be hardly any hardware components that we will have to directly manage since the Average Joe Classifier will be web-based and mostly software. The development cost of our project will mainly be the expense of keeping our database active for clients to use (software). We can rent server space for $100-$200 per month or buy a server upfront which will be approximately $1000-$3000 depending on the specific one we go with. For the website, we will purchase a domain name that we will name, "http://www.averagejoe.com", since it is easy to remember. The cost for the domain name will be $15 per year. The domain name will allow the user browsing on our website to access the website files stored on the web hosting server. Throughout the development, we may modify and tweak the product design that will lead to a change in our cost for software or hardware. We are budgeting for extra expenses. This project has an allotted amount of $1000 in case of emergencies or for extra expenses that come up during the development phase. In the future, after the software is implemented, there will be a need for a team to keep the website accurate, active, and refreshed. The general workforce will benefit from the production of this classifier because the maintenance work will allow for more employment opportunities. The cost of keeping the product active and worked on routinely will cost at least $22 per hour per employee, depending on what the position requires of that employee.

Additionally, the actual lines of code will be lengthy, and the files will be large. Since we are creating a database and using MongoDB for the backend of the classifier, we can estimate that the lines of code we write will be around 1,000-3,000. This code primarily corresponds to making the user's side of the website interact and communicate with the database. The code takes the user's photos and labels and then will be sent to more code to train and classify the object. After the classifier is trained, the information that the classifier has learned is sent and stored in the database. The data stored in the database will include large files with many lines of code, due to many users interacting with the website directly. The number of lines of code will be hard to predict for this since we do not know how many users will use the classifier and what files those users will store. The lines of code will vary due to the different storage amounts of files. MongoDB is a NoSQL database structure and the lines of code that it has are close to 2 million lines.

Furthermore, there will be a lot of milestones for this project. The first milestone will have a complete user interface in the form of a website for users to log onto. We need the website to allow a user to upload files of photos and/or videos that the user can then label and create bounding boxes around the object that he wants to classify. The website will display the information needed when the computer is learning an object. The website will show the results after the classifier trains on the images and allows the user to see where adjustments need to be made. The second milestone will be to have the website communicate with the machine learning code. With the machine learning code, YOLO will properly learn the object the user wants to classify. This is where the machine will correctly learn the object that the user gives it. This code will be the "brain" of the operation, which allows users to properly train classifiers. The third milestone will be storing the classifiers properly into the database. From here, the database will store the classifier if the user needs to refer back to it in the future. The database will need to know how and where to store the images and/or videos for the classifiers. A sub-milestone to this will be to incorporate the correct create, read, update and delete operational functionalities. This will allow the user to not only create and delete information from the database but also, correct image labels and update the classifier(s) when needed. For the fourth milestone, we need all three main parts of the classifier to communicate with each other to push out the user's desired classifier. This includes having the user's interface working and showing the user the results that the classifier has given back to the user. The machine learning code will learn further when the user adds, deletes, or modifies information for the database. To meet these milestones, we will start with communication with the three main components of the project; the three main parts are (1) the user website, (2) the YOLO algorithm that drives the machine learning, and (3) the database.

Over the next year, we will be pushing towards all of these milestones, however, it is hard to place a hard deadline for each of these milestones. This is because most of the milestones that we are working towards allow our team to concurrently work at an individual level for each. We will have many of the milestones started together and working partially by the end of this year (late December), instead of working on one milestone at a time. This way we can evaluate and predict how the pieces will fit together. Since we are adding a little bit of work to all of the milestones at once, we will not have to do too much extra work when figuring out how to integrate the pieces. If we decide to work on each milestone one at a time, we will not be able to understand the whole problem since we will be focused on the component we are working on. So, the timeline is loosely fit since we are working on each milestone at once and we aim to have 40% of the project working by the end of December, which means 40% of the milestones we want to finish will be complete by then. By the time January rolls around, 70% of the project will be complete which means that all the milestones are at 70% completion. We aim to have 90% of the project completed by February and will complete the project by the end of March 2020. In February, we will have 90% of the milestones completed and 100% of the milestones by March.