Machine Learning-Powered Image Processing: A Survey of Current Techniques and Future Directions

A survey conducted as a course requirement for CS 591 – Machine Learning

Tyler Conger
College of Engineering
The University of Alabama
Tuscaloosa, AL
tconger1@crimson.ua.edu

Patrick Vest
College of Engineering
The University of Alabama
Tuscaloosa, AL
pcvest@ua.edu

I. Introduction

Machine learning-powered image processing is the act of leveraging machine learning algorithms to perform tasks on images that would typically require human intervention or manual programming. This can include tasks such as image classification, object detection, image enhancement, and anomaly detection.

Machine learning-powered image processing is used in many different areas of industry every single day. In the medical field, machine learning can be used to analyze medical images such as X-rays, MRIs, and CT scans. In law enforcement, facial recognition and license plate readings help keep the public safe. Machine learning-powered image processing can be used by schools to grade handwritten quizzes or assignments for teachers, help consumers search for products online, help render realistic visual effects in film, and detect defects on manufacturing lines. All these uses only scratch the surface of the possible uses of machine learning-powered image processing.

To survey the applications of machine learning-powered image processing, we will start by examining CAPTCHAs.

CAPTCHA stands for *Completely Automated Public Turing test to tell Computer and Humans Apart*. Colloquially known as the lowercase *captcha*, they are tests that are used by websites in an attempt to determine if the user of the website is a human or some sort of malicious or non-malicious automated web crawler.

The purpose of a captcha is to prevent spam on websites. The first commercial use of the captcha was in the year 2000 when idrive.com used a captcha to protect its signup page.

The first version of the captcha required users to identify a sequence of distorted letters and/or numbers that a computer would not be able to identify with OCR, but a human could read with minimal effort.

In 2007, Google released its captcha service *reCAPTCHA*. ReCAPTCHA was a second version of the captcha, offering improvements over the original. Similar to the original captcha, reCAPTCHA was later improved to allow the captcha to only be displayed to the user after an analysis of the cookies on the device was performed. The third version and later of reCAPTCHA implemented a system to monitor user activity such as mouse movements and clicks and only displayed a captcha puzzle if the behavior on a page suggested that the page was being visited by a computer.

Webpage security is not the only purpose of captchas. Early captchas were also used to improve machine-learning models for Optical Character Recognition. They did this by showing a user two words – one known word and one word that OCR could not recognize. In displaying this pair to a user, captchas could crowdsource the characters that OCR could not recognize while keeping a website secure. Google used this system to finish the project of digitizing the archives of The New York Times.

Captchas are also used to inform the computer vision models that self-driving vehicles use. Figure 3.1 shows an image captcha from reCAPTCHA. In an example like this, the position of vehicles or other roadway information can be crowdsourced through the captcha system.

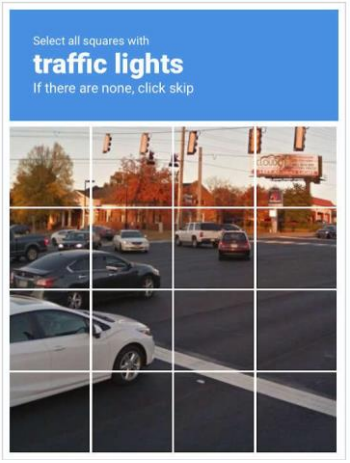

Figure 3.1 – An image reCAPTCHA

Google sells the data from hundreds of thousands of captchas solved every day to companies to use to train their computer vision models for self-driving vehicles.

Additionally, captchas are often used in combination with log-in security measures on websites. One such authentication method is facial recognition.

While it might seem like a stretch, the captcha is a great example of a simple product that informs facets of machine learning. In this research survey, we will analyze the areas of machine-learning-powered image processing that both directly and indirectly relate to captchas as well as these areas' future directions. We will explore uses in Optical Character Recognition, or OCR, facial recognition, and computer vision in self-driving vehicles.

While this survey will only cover machine learning topics related to, facial recognition, roadway analysis, and Optical character recognition, it is important to understand some other use cases of machine learning-based image processing tools. One such example of machine learning's use in image identification is that of medical imaging. In medicine, it can be extremely useful to have the secondary opinion of an artificial intelligence to see patterns that a human doctor may not see in certain MRI or CT scans. These tools can help to identify issues or potential things like cancer. At the very least they help to flag potential places where a closer look from a doctor would be very useful.

Some other potential use cases include use in manufacturing applications to identify defects, or in a warehouse or store to identify products remaining in stock. Many of the different algorithms we will discuss can be used in various applications and settings. Even still, only a small minority of the various algorithms and techniques available will be surveyed. It is worthwhile to understand that this will serve as a simple overview of information, because of the depth of information surrounding these topics. For each topic, the associated history, segmentation process, algorithm and performance comparison, challenges and bias, and the future directions of these algorithms will be discussed.

II. Facial Recognition

Facial recognition is the process through which an individual can be identified and matched to a previously existing photo in a database, given an image or a video stream of the subject. These images could come from a wide range of places including social media, CCTV cameras, governmental databases, or any number of other places. These individuals are matched to the stored database image or another image of them through facial recognition technologies where machine learning algorithms are used to identify and match the two photos to confirm the identity of the user of the system.

A. Introduction

What was once an idea of science fiction, facial recognition technology has become very prominent in our digital world. With an extremely wide range of applications - from security and policing and health care - to simple filters on social media applications, facial recognition has become ever-prominent and extremely important to understand. The use of facial recognition technologies has grown substantially in the past few years; however, it has long been used and studied. Today numerous different approaches and algorithms are used in the facial recognition process that will be discussed, each with their strengths and weaknesses. For example, some algorithms are better suited in specific situations like a side, or incomplete view of the face, whereas others may be better suited for a straight-on and clear view of the face.

B. History

Facial recognition technologies can be traced back to Woodrow Bledsoe, an early computer scientist and mathematician who, in the 1960s, worked on the development of a system to identify faces in a database. This early system worked by use of human input to directly measure the distance between distinctive features on the human face by hand, such as distance between the pupils, distance to the nose, and width of the mouth. [1]

However, while this sounds like an extremely rudimentary algorithm, this idea is often still used, without the human requirement of measurement, where computers can determine and detect these distinct points and measure distances between them, aided by the development of machine learning models.

As machine learning has helped to improve the accuracy, speed, and usefulness of facial recognition processes, they have expanded in use rapidly. Facial recognition is now used by many consumers to unlock their cellular phones and to take and edit photographs with facial filters. It also has uses in security situations such as airport security systems and criminal identification. As such, with the high number of potential benefits and uses, it has become a more deeply explored and studied topic in recent years as more private and public sectors have come to use facial recognition technologies.

C. Segmentation

The first step in the facial recognition process is to divide the face into individual segments. Once a face is separated into regions, feature identification can be done on those regions. In facial recognition, the process of identifying each feature is called principal component analysis (PCA). The first step of principal component analysis is region segmentation. Then, these individual regions are extracted. The quality of the image can affect the segmentation process and create issues if the image is of poor quality or resolution, hence a strong segmentation process is extremely important as it will inform the effectiveness of the algorithms that will be run on an image following the execution of principal component analysis. With this standard segmented data, what is called an eigenface is created which is a set of vectors that are derived from the face in computer vision. These vectors are derived from the features on the face and the depth of those features in 3D space. [2] An example of this standard eigen – a composite of how the computer interprets the input face that is detected – is shown in Figure 1.1 The figure displays the variation in depth as well as the major vertexes that emerge within the facial features of the subject. While these images appear to be blurry and unclear, they represent only the first step in the image processing algorithm.

Figure 1.1 - Examples of Eigenfaces as interpreted by the machine learning algorithm. [3]

Another commonly used technique is SIFT. During the process of segmentation, Scale-Invariant Feature Transform, or SIFT, for short, is often used due to its robust handling of facial appearance variability and helps to maintain accuracy. It is usually applied to the input faces before they are compared to a reference image. These SIFT features are then extracted and compared to help with the image-matching process in a later step. SIFT identifies the area of interest that is used for proper identification and comparison between the two images. [2] In the image below we can see images with the SIFT identification of features. We can see the highlighted places in the image and the corresponding feature in the eigenface-processed image on the right. The important features are extracted from the face on the left side and are mapped to matching features on the right image. This allows proper recognition of a like face between the two separate images.

Fig 1.2 Example of SIFT extracted features showing the correlations between the original image and a processed eigen face image. [2]

A machine learning process can be used to improve segmentation as image processing can be limited by things like low camera quality, insufficient viewing angle, pose, lighting conditions, and wearing accessories like masks or sunglasses. This early identification is vital to better preparing detection algorithms for the next step of the process. These segmentation algorithms must continue to improve as that will accelerate the development and accuracy of the facial feature comparison steps.

D. Algorithms

There are a few algorithms that are used in facial recognition this paper will touch on: Linear Discriminate Analysis (LDA), Support Vector Machine (SVM), and K nearest neighbors (KNN). Each of these algorithms is slightly different from one another but still accomplishes the task of facial recognition. The actual facial recognition occurs after PCA has already been completed and the image has been appropriately segmented. In general, most facial recognition algorithms are, in some sense, minimum distance and thus the way that the distance is measured and interpreted is the most important feature of these types of algorithms. [3]

First, the LDA methodology makes use of separating the data mathematically. It uses discriminatory analysis to identify features. These features function by analyzing pixel values. Each is classified into distinct categories, such as texture features and shape features. [4]

Another algorithm used in facial recognition is SVM. SVM is an algorithm that can provide additional dimensionality and pattern recognition abilities. When using SVM, it is important to delineate the difference in the image between the point surface of the face and the decision surface. Points are chosen on the decision surface and then a distance is generated between this point and the decision surface, which is then known as the support vector. The closest point to this is known as the support vector point. The margin of width, the distance between points, and the distance to the nearest point are used to detect similarities between facial images and thus identify the face. [4]

Another type of algorithm used is the K-nearest neighbors algorithm. This is sometimes considered a significantly slower algorithm as it must run K times and thus it can be more computationally complex, depending on the number of runs that are taken. In the KNN system, an object is classified by a "vote" from its closest neighbors. Then boundaries are established between these separated groups that allow for classification among the images. When features are placed similarly between images it increases the probability the images are of the same face. [2]

Similar to the eigenface it is important to understand how we determine the facial features that are measured. Some differing algorithms for this include elastic matching and the use of neural networks. Elastic matching is better suited for situations in which the compared face looks different or displays a different expression to the original, such as a person smiling in one photo and not in the other. With strict comparison of distance between points, some matching may struggle, but elastic matching is much better suited to this scenario. [3] This is because it allows for matching in a situation where the values do not need to be as exact but can instead have slight variation to maintain a better matching rate, even if slight differences between each exist.

Comparatively, a geometric-based method may be better suited to images that have the same facial expression, photo angle, and quality, like a mugshot or police database. In geometric-based algorithms, each facial feature's location, such as lips, eyes, mouth, and nose, and the characteristic vectors associated with each facial feature are recorded. Then each of these facial feature positions are compared with the control image allowing facial recognition to occur in faces that have the same make-up. As the location of facial features cannot be changed, a face is like a fingerprint where the features are unique to each individual, and features cannot be replicated or modified. [4] However, this does require photos that maintain the same pose as an expression could change the precise location of the features and make the detection process more difficult.

E. Challenges and Bias

Facial recognition processes have come into criticism in the past for major bias issues, some of which are attributable to data training sets, while others are more deeply rooted in the inefficiencies of the algorithms themselves. This is especially important as facial recognition is used by police departments and government agencies, where mistakes can be extremely costly and permanently damage lives. As previously discussed, training data can have a strong effect on convolutional neural networks and the resultant data, as such data with an initial bias towards or away from one race or gender can have a devastating effect on the later algorithm causing issues.

According to Alex Najibi of Harvard University, there is up to a 34.4% difference in accuracy between light-skinned males and dark-skinned females. This kind of discrepancy shows an enormous difference between the two in detection ability. Across four different facial recognition technologies, from Microsoft, Face+++, IBM, Amazon, and Kairos, with accuracy ranging from as low as 20.8% difference to as high as 34.4% difference. Seeing the lowest discrepancy of 20.8% shows that many of these popularly used algorithms possess flaws. This data disparity can also be seen more clearly in Table 1.1 below. Najibi also cites that facial recognition is used to target marginalized population sectors such as undocumented immigrants and Muslim citizens. Najibi also states that black individuals are more prevalent in mugshot databases, and thus when facial recognition software makes predictions pulled from this data, the result disproportionately discriminates against and leads to the arrest of more black individuals. [5] We are also able to see that these disparities are not just racially based, but also vary between genders - with females having generally less accuracy when compared to their male counterparts' males when using these companies' facial recognition technologies.

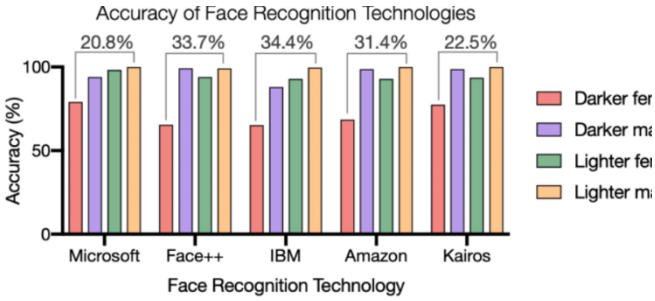

Table 1.1. Data visualization displays the varied accuracy of each company's facial detection software and the vast disparities between different races and genders. [5]

These issues are major and need to be addressed to better level the playing field and improve the overall efficacy of facial recognition technologies. Addressing them would include both updating and improving upon algorithms as well as improving upon the initial datasets that these algorithms are trained on to better encompass the true diversity of the world's population. Both aforementioned factors can have a great effect on the accuracy of the overall algorithm and reduce discrimination within facial recognition.

F. Future Directions

Facial recognition has already proven its usefulness in many different private and public sector applications. Now, as we continue to hone the algorithms and improve them over time, we can expand the number of situations in which facial recognition is used. All the challenges that include discrimination between genders and races with facial recognition technologies must be addressed to help prevent issues and bias against individuals of color and gaps in algorithmic performance between race and gender. It is extremely important to address challenges with darkness and shading, especially on darker versus lighter faces. As well it is also important to address differences in gender and accurate identification. Part of this can be solved with improved data sets that are less biased and contain more information, but it also requires that the algorithms themselves be better at detecting darker faces or faces with masks.

One of the major drawbacks of more powerful facial recognition technology is the processing power required. As both processing power gets better and the algorithms behind facial recognition become more refined, we will be able to see a lower barrier to entry on these high-performance facial recognition algorithms, allowing for their use in more places. As the technology improves, we will begin to see it used in a greater number of situations such as banking, security applications, and even concert tickets. There is a true potential for facial recognition to be applied everywhere.

G. Conclusion

Facial recognition technologies have come a long way from the original hand-measured distance of key features to help in the identification of individuals, moving toward new machine learning algorithms with various levels of speed and accuracy. It is important to keep in mind that more improvements both to the data sets and the algorithms themselves are necessary to better service the individuals that use them. [4] As these algorithms and data sets improve so too will the results including the speed and accuracy of facial recognition technology.

III. Roadway Analysis with Image Detection

Roadway analysis is the process of using visual data to identify specific obstacles on the roadway. These types of obstacles can be vehicles, roadway obstacles, and even objects on the side of the road such as road signs. Each of these is extremely important to self-driving vehicles, especially as this is an increasingly emergent field. It is also important in other fields such as roadway maintenance and traffic management and analysis. This topic will cover various algorithms and processes of analyzing images and camera footage to identify items such as road signs and roadway obstacles, a process in which efficiency and accuracy are extremely important. Many of these algorithms must operate in real-time at high speeds around humans, where a failure could create an extremely dangerous situation.

A. Introduction

Analyzing and understanding the roadway ahead is an extremely important part and a key component in modern-day vehicles. Features such as driver assist, and roadway obstacle notifications are becoming more and more common in newly produced vehicles – even without fully self-driving capabilities. The necessity of these safety features is especially prevalent in self-driving vehicles and more automated driving solutions. The use of image processing can be used to visualize road signs, roadway conditions, other vehicles, and traffic conditions. These vehicles and cameras can gather thousands of hours of video footage and photos, which machine learning algorithms can interpret extremely fast - allowing for real-time analysis in the case of self-driving cars.

Image analysis can be used to identify the boundaries of the roadway itself, while also classifying details such as lines painted on the road as well as potholes, cracks, and other types of damage or obstacles. This can be useful in mitigating accidents and potential damage to the vehicle. These vision-based detection systems are implemented in some modern vehicles with technologies like lane assistance and accident avoidance. Detection of roadway hazards can be attributed to some common algorithms such as Region-based Convolutional Neural Networks (R-CNN), single-shot detection (SSD), and You Only Look Once (YOLO). These algorithms will be discussed further in the algorithm analysis section; however, it is important to know that all of these discussed algorithms are used to identify and detect obstacles in the roadway. These types of identification also include obstacles such as distances to other cars and the distances to the roadway boundaries such as curbs or edge barriers. Additionally, in the case of autonomous vehicles, it is important to understand other peripheral roadway information such as the signage alongside the road.

Image analysis can also be used to identify signage along roadways, such as posted speed limits, traffic lights, stop signs, and other roadway safety signs. This allows an autonomous vehicle to understand the rules of the road and comply with safety standards. For this type of identification and processing, algorithms like color segmentation can be used to identify signage, but other algorithms like Single Pixel Voting (SPV) can be used to identify sign color and information. Another

technique such as Contour Fitting (CF) can also be used. Contour Fitting is an algorithm that can be used to identify the size and shape of signs and determine the type of sign. Another technique that is commonly used is Pair-Wise Pixel Voting (PWPV), which unlike the others does not require color segmentation, and uses Hough Transformations instead. Now it is important to cover and understand each one of these different algorithms and the associated processes with them in a greater degree of detail and understand how machine learning pertains to each one.

B. Segmentation

The first step before any of these algorithms can take place is segmentation. This important first step takes the image that is provided and breaks it down into individual segments that can be used. An example of this is taking the provided image and dividing it into a grid of sections. Each of these sections is analyzed and processed by the system to result in faster run times and a more complete identification algorithm. These sections allow for empty sections to be discarded and not used. As we can see in Figure 2.2, only sections in which there are existing objects that are detected are analyzed and the rest are removed.

This process of segmentation is especially important in roadway identification, as it decreases processing time, which can be important in real-time applications. Video frames are processed by the machine and divided into a grid and each individualized grid section is then analyzed later with the algorithm to identify objects in the roadway, signage, and other vehicles. This allows for the identification of an item in space as we know what section of the grid it is in, thus its "coordinates" and then the item can be properly identified and the response to that obstacle can be determined.

Another form of segmentation that is commonly used in roadway analysis such as roadway signage detection is color segmentation which can be then followed by a recognition stage which may use various types of algorithmic approaches. While these approaches vary, the optimization method is to shift the image to greyscale excluding a few specific colors, such as bright reds, bright blues, or another specific color. This shift allows for colored signage to be more easily identified. Thus, a red stop sign is much easier to detect and categorize than it would be otherwise. Oftentimes times these grey-scaled images can also be reduced to grids as previously described to better identify any roadway signs in the image. [6]

C. Classification

As for some of the algorithms used in terms of distance and obstacle detection, some common ones as discussed previously are R-CNN, SSD, and YOLO. These machine learning and deep learning algorithms are extremely important in image-based approaches and use cases such as self-driving vehicles. The first of these algorithms, R-CNN is a version of CNN that can be used on 2D images for analysis. In CNN, various digital filters are used at various stages to capture information from the image, which is then subsampled to collect data about each filtered class. R-CNN is a similar model to CNN that uses regions to allow faster object detection. R-CNN uses three modules. The first module generates regions, the second extracts feature vectors from each of the regions, and the third contains linear support vector machines. [7]

It is worth noting that there is a large variety of different R-CNN and Convolution neural network techniques used in roadway analysis. Some of these are optimized algorithms that allow for uses in higher-speed situations where data is needed much more quickly and thus some sacrifice to accuracy must be made. This includes examples like R-CNN fast and faster R-CNN which are both used and as their names describe are faster variants with tweaked algorithms allowing for more efficient calculations and uses. However, this discussion will just focus on a singular variation of the CNN algorithm that is frequently used.

In single-shot detection (SSD) an image goes through convolutions for feature extraction, where a feature map is obtained. Then each object gets a bounding box that is overlayed onto the grid created. Then, each object is given a class score as to how likely it belongs to a certain group of things, such as a traffic light, sign, pedestrian, or car. Then each object bounding box is given an offset to help it better align with the item in question. Two different types of SSD models are commonly used, SSD300 and SSD512, 300 is used for lower resolution but computes faster and 512 is more accurate but slower demonstrating a tradeoff between the two types of models. [7]

Another commonly used algorithm is the You Only Look Once (YOLO) algorithm, where an image is divided into a grid similar to SSD. In YOLO, grid anchor boxes are adjusted to fix each individually detected object. In the next step, a convolutional neural network (CNN) is used to process the image as a whole, capture features, and extract them from the objects in question. YOLO then uses non-max suppression (NMS) which reduces overlap and intersection between bounding boxes. One main advantage of YOLO is its speed in object detection. It is significantly faster than R-CNN. YOLO does, however, struggle on smaller objects, and items like road cracks or potholes may be difficult for it to detect and another algorithm may be necessary in these cases. The process of the YOLO algorithm can be seen in Figure 3.2 below. The image is segmented into different sections represented by the red grid boxes. The excess grid boxes that would have been along the top row have been discarded as they did not contain any detected objects. Then the detected objects are highlighted in blue. In this case, the objects detected are three soccer players and a soccer ball. The yellow stars represent the average center of these points and the average center of the associated object.

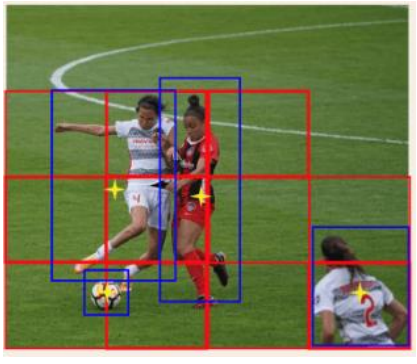

Fig. 3.2 Showcasing the YOLO object detection algorithm, showing segmentation of the image and object detection within the image. [8]

As also discussed, image processing can also be extremely useful in determining more information about the roadway such as a sign, while CNN and SSD can have some use in these cases too, other algorithms are also used. One of the most common is color segmentation which can be used as an initial processing mechanism of the image to identify certain colors in the image that are contained in roadway signs for faster identification. Another approach is the Hough Transformation (HT) which relies on gathering the edges of objects and identifying certain specific shapes such as circular roadway signs. This can be used in conjunction with color segmentation to help identify road signs. This works specifically well on circular speed signs or other circular signs as well. There are three major types of approaches to solving this problem first Single Pixel Voting (SPV) which is a system similar system to the Hough Transform. [6]

For SPV an image may be classified into each of the types of pixels such as red and not red pixels and blue and not blue pixels. This highlights objects of a single color in an image, such as the uniform red border of a sign. With these updated images, we can detect geometric patterns by detecting the border of a road sign and creating a bounding box around the sign. The Hough Transform algorithm, which is a type of the SPV algorithm, can then be used to process the image and create the appropriate bounding box. Then the geometric patterns are verified, and the image is processed appropriately. [6]

A secondary method that can be used is contour fitting (CF). CF uses a similar method of selecting red and blue pixels and building a map out of them, but also detecting edge points of the sign and using that information to fit it to rectangular, circular, or ellipse shapes. As we can see in Figure 3.1 below, the various signs have distinct shapes that allow them to be easily detected within the image using the CF algorithm. It can be seen in the larger image that all the signs are highlighted with their shape in either red for circles and ovals or green for triangular signs. Each of the detected signs is displayed at the top of the figure.

Fig 3.1. An example of using contour fitting to detect road signs within an image. [6]

Another commonly used algorithm, similar to single-pixel voting, is pairwise pixel voting. This method, however, unlike the previously discussed, does not segment based on colors. Then, the general process of this algorithm is that a transformation is used to detect the shapes of roadway signs. Then, for each of the edge points, based on the color and orientation relationship, a vote is cased for the middle point. Commonly used algorithms for this transformation include algorithms like Bilateral Chinese Transform (BCT) and Vertex and Bisector Transformation (VBT) The algorithms also identify the shapes - which are then "voted" on by each of the vertex to identify the type of signage detected. [6]

While understanding how each of the algorithms works, it is also important to understand the different situations in which each algorithm is used, as well as the pros and cons of using each algorithm, as each may be better versed for a certain specific situation. This is especially relevant when considering the computing time necessary to run the algorithm or the hardware necessary for the execution of the algorithm. As such, the performance characteristics are important to understand for the usage of the appropriate algorithm in each situation.

D. Performance

Each methodology used for object detection and sign detection has its strengths and weaknesses with some being more widely used in certain situations and others with a more nuanced approach. In the realm of self-driving vehicles, compute-time and compute-power requirements are extremely important as self-driving vehicles constantly capture and analyze information about their surrounding world. However, a slower more computationally intensive algorithm may still have a use case in a non-fully autonomous environment. Thus, it is important to acknowledge and understand the steps involved in each algorithm to better understand the processing and run-time complexities associated with each.

First, we will address the obstacle detection algorithms as discussed above. As mentioned, YOLO is extremely popular and widely used due to its comparably fast speeds and low compute times.

For SSD, one advantage that can be seen is that it only requires the use of a single network, and it operates faster than R-CNN does. However, SSD suffers when it comes to detection accuracy in comparison to R-CNN. Also, in comparison to YOLO, SSD is generally slower, but it is faster than R-CNN.

In comparison, the R-CNN method allows for more exact results - giving a more accurate location of detected objects. However, R-CNN is generally considered to be not fast enough, even in its faster form, to operate in real-time applications such as self-driving vehicles and thus is reserved for less time-sensitive applications. It is again important to note that there are different variations of R-CNN, however even Faster R-CNN follows these principles.

YOLO is best used in real-time applications such as autonomous vehicles as the location of objects has fairly strong accuracy, but the algorithm also has lower compute time, allowing for more real-world use. Unfortunately, this approach does have some difficulties with smaller objects, and thus things like cracks, potholes, or small objects in the roadway would be much harder to detect with any amount of accuracy and these small objects may be missed by the algorithm.

It is also worth discussing the advantages and disadvantages of image analysis algorithms that are used in the visual analysis of the roadway. The type of algorithm that is selected for a task is extremely important as smaller road signs may be more difficult to detect and thus the use of an algorithm that works on smaller signs is important. As such accuracy should be important here, because missing a roadway sign, even if it is

small, could cause the need for driver intervention in the vehicle's operation.

Algorithms like SPV and CF are nearly equally powerful, and both are subject to over and under-segmentation which can be damaging to the algorithm. However, with smaller signs, SPV is generally more accurate which leads to its use over CF. In lower compute situations CF may be more useful than SPV as it requires less compute resources for more accurate predictions or equally accurate predictions. It is important to note that when a color filter is used, the required computation time decreases dramatically. However, PWPV is operationally the faster and more effective algorithm with a generally much higher detection accuracy and lower detection errors. [6]

E. Challenges and Bias

While it is important to note that there are some sources of bias in roadway analysis, in this field of machine learning and image processing there are considered to be fewer sources of bias. Bias can come from both the training data and the resultant models than other fields of image processing such as facial recognition.

A substantial piece of the bias that is often discussed is that of pedestrian recognition, a process that is extremely important in self-driving vehicles. Data suggests that these algorithms struggle with identifying children and individuals of color when detecting pedestrians. There is a much smaller gap between gender identification, with systems nearly 20% more likely to identify and detect adult individuals than children. This is a huge discrepancy - especially in detecting members of an extremely vulnerable group, who often are less aware or less understanding of vehicles, it is important to be able to properly identify and properly respond to children in the roadway. Also, individuals of color were nearly 8% less likely to be identified than lighter-skinned individuals with even more discrepancy in low-light or low-brightness situations. [9] This shows how discrimination and bias play a part in roadway analysis and self-driving vehicles which could be extremely dangerous to these vulnerable population groups, and how that bias should be corrected swiftly and appropriately.

F. Future Directions

As we continue towards higher degrees of vehicle autonomy, it is important to understand the future goals and challenges. First, it is important to understand the levels of self-driving vehicles as described by the Society of Automotive Engineers, this information can be found in Table 3.1 below.

Level Number	Level Characteristics
Level 0 – No automation	No automation whatsoever. The driver always has complete control over the vehicle.
Level 1 – Driver Assistance	Extremely low vehicle input. Features such as cruise control. A singular automated system.
Level 2 – Partial Automation	The vehicle has a low level of automation but constantly requires the driver's involvement. An example of this is that it can perform basic tasks like steering and power control.
Level 3 – Conditional Automation	The vehicle can perform the majority of the driving tasks; however, driver attention and involvement are still necessary.
Level 4 – High Automation	The vehicle can perform all driving tasks, however under specific circumstances driver input may still be necessary but limited
Level 5 – Fully autonomous	There is no human input or attention required whatsoever. The car has all the functionalities and abilities that a human driver would have without any of the issues.

Table 3.1 The distinct levels of self-driving automation and key characteristics associated with each level.

As shown in the table above, we still have a long way to go before level-five autonomous vehicles are everywhere. However, we are developing vehicles that are increasingly closer to level 5. Looking forward, future companies that have reached level 3 autonomy are moving forward toward higher levels. Companies like Waymo already operate and have self-driving taxis on the roadways. The number of self-driving cars in use will only continue to increase with better computing hardware and more optimized algorithms.

As previously discussed, algorithms and machine learning techniques are continually refined. These improvements will only help in reaching higher levels of autonomy in vehicles. As described in Table 3.1, with higher levels of autonomy less driver intervention is necessary, which decreases the risk of human error or a traffic accident occurring. It is also important to understand that the datasets associated with roadway analysis will continue to grow with increased autonomous vehicles on the roadway. The vehicles will continue to become even better and improve in safety and function as the algorithms are essentially crowdsourced by their deployment, resulting in better navigation on our roadways.

G. Conclusion

Overall, there are many situations in which image processing is used in roadway analysis, and depending on the situational need, differing algorithms can present different use cases. In a situation where time sensitivity and compute power necessary are extremely important a more efficient, less nuanced algorithm should be used. Understanding and continuing to improve these algorithms, how they work, and what they can be used for is important in determining which algorithms may require further in-depth research and which algorithms should be phased out in favor of future developments.

In a self-driving vehicle when a given scenario is extremely time-sensitive, it would be best to use the YOLO algorithm as it would provide the best real-time data with the most accuracy. As well as using a separate algorithm for sign detection such as SPV or CF - which could be used depending on the available

hardware.

As for the bias that is associated with each of the different algorithms used, the largest bias issue may depend on signage type, different countries and regions use distinctly different types of signage some of which are much more difficult for SPV or CR to read. An example of this would be in Europe, a smaller triangular pedestrian sign may be more difficult to detect than its larger counterpart in the United States. While self-driving vehicles have become more and more mainstream, it is important to understand all of the safety features along with these vehicles as well as make sure these vehicles will operate without fail in a multitude of different environments. These changing environments include driving between countries, where sign types, and roadway rules may not be completely standardized across borders.

IV. Optical Character Recognition (OCR)

Optical Character Recognition or OCR is the identification of printed characters using photoelectric devices and computer software. Simply put, Optical Character Recognition is the action of recognizing printed or handwritten text and converting it to or generating a copy of the text in the form of digital plaintext.

A. History and Practical Applications

Although the original concept of Optical Character Recognition can be traced back to the early 1900s, Optical Character Recognition computer systems only began to emerge in the 1960s and 1970s. Originally, Optical Character Recognition was used for niche use cases, such as recognizing text in handwritten notes or reading the zip codes from mailing envelopes. At the time, Optical Character Recognition was limited to a small subset of typed fonts and was not extremely accurate. Due to advances in machine learning, Optical Character Recognition would soon become more widely adopted by consumers for daily tasks.

By the 1990s, Optical Character Recognition systems began being used more widely on personal computers for digitizing printed media such as receipts, books, and magazines. [10]

Today, Optical Character Recognition is used by consumers on mobile phones to digitize receipts and documents, and even aid in real-time translation of printed media using products such as Google Lens. Which allows for real-time translation of printed text to a digitized version. Optical Character Recognition is also used by speed cameras to capture license plate characters and self-driving vehicles to recognize road markings and signs, as previously discussed in section II.

B. Segmentation

For OCR to be effective, an Optical Character Recognition system must execute a series of steps. The first step is preprocessing, which includes converting the image to grayscale, denoising, etc. – then segmentation and identification.

Segmentation is the process of separating a dataset – in the case of Optical Character Recognition, an image of text – into logical partitions of similar information. For an Optical Character Recognition system, these partitions are the letters or distinct symbols that make up a string of information. There are many popular segmentation algorithms from many different vendors. The algorithms analyzed in this survey are among the most popular and accurate Optical Character Recognition Algorithms in use today.

i. The Tesseract Engine and Connected Component Analysis

One popular segmentation and identification algorithm is The Tesseract Engine. The Tesseract OCR engine was originally developed as a Ph.D. research project in Hewlett Packard Labs and first appeared at the University of Nevada – Los Vegas (UNLV) Annual Test of OCR Accuracy forum in 1995. The Tesseract engine proved to be an exceptionally reliable engine at the conference, boasting accuracy scores of over 95% in most categories. [11] However, after the Annual Test of OCR Accuracy, the Tesseract Engine seemed to disappear from the public eye, but it is still a powerhouse in the Optical Character Recognition field today.

The Tesseract Engine operates in a series of steps that it uses to segment and identify characters. The first step is connected component analysis. Connected component analysis created outlines of similarly colored groups of pixels that are likely to be a letter or part of a letter. The connected component analysis was historically computationally expensive, but modem processors can accomplish this step with ease. In performing the analysis, the Tesseract engine can easily recognize markings inside other markings despite the overall color of the image. This feature made the Tesseract Engine one of the first OCR engines that was able to easily process black text on a white page *or* white text on a black page. At the end of the connected component analysis phase, the markings, along with their child marking – markings inside others – are grouped in *Blobs*. [11]

The *Blobs* are then organized. The Tesseract engine organizes the blocks into lines and then into either words or regions to perform pitch and skew adjustments in a localized manner. This step, usually done in the pre-processing phase – before segmentation, is done at this point to allow the engine to have a precise level of control over the preprocessing of a region of text as opposed to an entire document or page.

After the blobs are gathered and processed, the individual words, as defined by the spacing between characters, among other factors, are recognized. If the engine determines that a word has been recognized satisfactorily, the engine passes the recognized word on a system called the Adaptive Learning Classifier.

The Adaptive Learning Classifier allows the engine to use previously satisfactorily recognized words in comparison to words further down the page that may be difficult to recognize. This case-by-case learning model allows the Tesseract Engine to become more accurate during an individual use as opposed to the entirety of a use being used to train future uses.

In the final step, after an entire page or document has been processed, the Tesseract Engine uses the data that it collected from the start of the use to process *Fuzzy Spaces*. Fuzzy spaces are spaces between words or special characters that the Engine did not satisfactorily identify in the first pass. Using the data from the rest of the document, the engine can resolve the fuzzy spaces and complete the Optical Character Recognition

process.

ii. Document Layout Analysis and Region Proposal

The segmentation of characters in a document is crucial to the success of Optical Character Recognition. However, what about the segmentation of fields in a non-linear document such as a receipt? Here, a new technique becomes necessary: document layout analysis. Document layout analysis is a technique employed by many computer vision engines, including Google Cloud Vision which is exceptionally good at recognizing handwriting.

The document layout analysis process has *many* steps. Far too many to fit in a short section of this paper. However, two main steps make use of a technique called region proposal and are therefore relevant to the study of Optical Character Recognition: Text Block Detection and Word and Character Segmentation.

Region proposal is the process of an Optical Character Recognition system analyzing a page of information and proposing a set of bounding boxes that likely contain objects of interest passing it to the next phase of the Optical Character Recognition Engine.

Unlike the Tesseract Engine, which performs the character recognition process on a linear progression of tokens, region proposal aims to determine what information in a document is useful and group that information in logical *regions* for processing. [12] Take a receipt, for example. On a typical receipt, line items' prices are listed in line with the items' names. Despite this easily human-readable layout, a user may wish to extract text from the item names column and the prices column separately if copying the text to a spreadsheet. Region proposal allowed for this type of copying but segmenting a page into regions before performing the textual analysis.

Once the region processing is complete, the *Regions of Interest* or ROIs are passed to the prediction head, also called prediction branches. The prediction head is a set of neural networks that are trained to use predictive geometry to refine ROI bounds and then classify pixels in the image as part of a specific character or object on the page. The per-region prediction engine allows engines using document layout analysis to recognize regions of text that overlap, making the engine much more efficient at recognizing handwriting or fonts with varying tracking, or spacing, between characters.

C. Performance

Optical Character Recognition is not perfect. That is why there are many Optical Character Recognition techniques. Each technique can vary in its usage of segmentation and classification algorithms, which can vary in their efficacy.

In a 2015 study by Dr. S. Vijayarani and Ms. A. Sakila, 8 Optical Character Recognition tools were compared. The 8 tools were tested with sample data containing plaintext and special symbols such as Greek letters and mathematical operators. The test results showed that the most popular OCR algorithms were extremely accurate with plaintext but struggled tremendously with special characters, where all the OCR tools from the table scored a special symbol error rate of 100%. [13] The [truncated] results from the study are shown in Table 1.1.

S. No	OCR Tool	Character Accuracy (CA) (%)	Special Symbols Accuracy (EA) (%)
1	**Online OCR**	**95.9**	**0**
2	**Free Online OCR**	**98.64**	**0**
3	**OCR Convert**	**100**	**0**
4	**Convert image to text.net**	**100**	**0**
5	**Free OCR**	**100**	**0**
6	**i2OCR**	**100**	**0**
7	**Free OCR to Word Convert**	**23.29**	**0**
8	**Google Docs**	**100**	**0**

Table 1.1- Results from "Performance Comparison of OCR Tools" (2015)

D. Challenges and Biases

Though modern Optical Character Recognition engines are proven to be very accurate, they can and do exhibit biases or confusion – especially those that make use of machine learning.

The most common letter bias, by far, in the field of Optical Character Recognition is the *errant 'e'*. 'e' is the most commonly used letter in the English alphabet; therefore, many OCR engines will be slightly biased to a character being an 'e' rather than a 'c' if the distinction is not clear.

In an annual research report from the Information Science Research Institute, researchers noted the most common OCR confusions. The 10 most common confusions are listed in Table 1.2 below.

	Correct	Generated
1	[blank]	space
2	e	c
3	,	.
4	space	[blank]
5	1	l
6	i	l
7	O	0
8	0	O
9	1	I
10	a	s

Table 1.2 - The 10 most common OCR confusions [14]

Another common challenge that OCR engines face is *with* handwritten text. While modern OCR systems such as Google Cloud Vision OCR are capable of recognizing handwritten text, the process is far from perfect. Handwritten text formatting not only varies between samples but can also vary between the same letters in a single sample. Humans may write an 'e' that follows an 'h' different than an 'e' that follows a lowercase 'L'. Because handwriting samples can vary so wildly, the training data needed to build a handwriting OCR engine is enormous. This can make handwriting OCR engines prohibitively expensive monetarily and computationally.

Though handwriting is difficult to perform OCR on, OCR systems built on the Tesseract Engine have shown promise. Because of the adaptive learning classifier built into the Tesseract Engine, handwritten text is recognized better throughout a sample.

E. Future Directions of OCR Technology

In the ever-evolving landscape of technology, Optical Character Recognition (OCR) stands at the brink of a transformative era. The challenges and biases of the past, including biases toward certain characters and struggles with handwritten text, are now becoming catalysts for innovation. A look to the future shows that OCR technology is poised to make significant leaps in accuracy, versatility, and applicability across various domains.

i. Accuracy-Improving Techniques

Humanity's pursuit of higher-accuracy computing continues to drive the development of innovative OCR techniques. Machine learning, deep learning, and neural networks are at the forefront of this revolution. The result of the use of these technologies is a reduction in errors, particularly when dealing with complex fonts and scripts. OCR systems are becoming smarter, evolving to comprehend and adapt to intricate details of fonts and handwritten text.

ii. Multilingual OCR

As globalization continues to the world, the demand for OCR systems that transcend language barriers is growing. The future of OCR technology will likely involve improved multilingual support. This means that OCR systems will seamlessly extract text from documents in various languages, facilitating international business, research, and communication. Google Cloud Vision API is an excellent example of this technology of the future in use today. Google Lens has support for near real-time OCR and translation of signs, menus, and other foreign media.

iii. Handwriting Recognition Advancements

As mentioned previously, handwriting recognition remains a challenge for OCR, but it is a challenge that develops are actively addressing. Advanced machine learning models, neural networks, and artificial intelligence coupled with access to practically unlimited datasets of handwritten text, are essential to the future of handwriting recognition. These advances are reshaping OCR's capacity to accurately decipher and transcribe handwritten text. When used with multilingual OCR, these advancements could soon make manual data entry and translation a thing of the past.

F. Conclusion

Optical character recognition is not perfect. However, the use of machine learning in both pre-processing and the recognition process has helped OCR become a tool that is widely used by consumers and industry professionals alike. Machine learning has allowed OCR engines to learn on the fly with adaptive classifiers and predict regions of text outside of standard line rules. As computer vision processors and graphical processing units become increasingly powerful, OCR will surely become more powerful along with them. With many advancements in OCR development on the horizon, particularly from Google, OCR stands to improve the lives of consumers to a greater degree. With higher accuracy, multi-language support, and the use of artificial intelligence and neural networks, OCR could one day reach a point where it functions better than a human eye. That may seem like a bold prediction, but computers are already capable of mathematical operations, sorting, and video editing faster than humans. OCR could very well be next.

V. Conclusion

In this survey, we have examined the pivotal role of machine learning in image processing across three crucial domains: roadway analysis, facial recognition, and OCR. Our exploration revealed remarkable advancements in classification and segmentation algorithms such as principal component analysis in facial detection, R-CNN in roadway analysis, and document layout analysis in OCR. Our survey also highlighted the practical applications of these technologies in their respective industries. We have delved into the performance considerations, technical constraints, as well as the multifaceted challenges and biases that shape the landscape of machine-learning-powered image processing.

Machine learning-powered image processing is vastly important to the future of computing. Those in the field of machine learning need to understand both the history of machine learning-powered image processing and its future directions so that they may have clear visions and goals as they continue to make contributions to the field.

In this survey, we discussed many of the environmental and technical constraints of machine learning algorithms. Many of these constraints come in the form of computational efficiency. As machine-learning algorithms become increasingly complex, more robust computing hardware is required to run calculations. Additionally, packing increasingly powerful hardware into smaller spaces is becoming increasingly necessary as self-driving vehicles and drones attempt to shed weight.

The machine learning-powered image processing domains discussed in this are not without fault. The most egregious of the challenges and biases that we covered; the challenge of detecting children with roadway analysis, serves as a reminder to all in the field of machine learning that there is always a way and a need to improve the reliability of image processing.

Through the in-depth study of each domain covered in this survey, we also talked about the future directions of machine learning as it relates to the domain. In the field of facial recognition, the barriers to entry for the practical use of facial recognition for consumers may soon be no more. In the field of roadway vision, the future will see self-driving vehicles require continually lesser human interaction. Lastly, OCR will sooner be able to properly recognize handwriting better and even handwriting in other languages.

The insights gleaned from this survey underscore the pivotal role of machine learning in image processing and the vast potential that is held for many industries. As we as computing professionals, researchers, and policymakers consider both the societal impact and technological innovation of machine learning, we must take some crucial steps.

Firstly, we must sustain research and development and

expand the frontiers of image processing. This includes improving existing algorithms and devising new solutions that address current limitations and biases.

Secondly, we must use machine learning responsibly. Especially when machine learning-powered image processing may be used in some law enforcement processes, safeguards must be set to ensure that the public is treated fairly and in an unbiased fashion.

Lastly, we must stay informed. Given the rapidly evolving nature of technology, researchers and developers must continually stay up to date on new developments. Through education, we become better prepared to make new advancements.

In conclusion, the journey through the landscape of machine learning-powered image processing has underscored the remarkable achievements of human ingenuity and the necessity for careful management of these potent technologies. We must keep in mind that there are challenges on the way as we proceed. However, it is precisely by surmounting these challenges that we will further elevate the role of machine learning in shaping our future.

VI. Bibliography

[1] J. Ehrenfeld and C. Libby, "Facial Recognition Technology in 2021: Masks, Bias, and the Future," *Journal of Medical Systems,* p. 3, 2021.

[2] P. KAMENCAY, M. ZACHARIASOVA, R. HUDEC, R. JARINA, M. BENCO and J. HLUBIK, "A Novel Approach to Face Recognition using Image Segmentation Based on SPCA-KNN Method," *Radioengineering,* 2013.

[3] J. Zhang, Y. Yan and M. Lades, "Face Recognition: Eigenface, Elastic Matching, and Neural Nets," *IEEE,* p. 13.

[4] P. Singhal, P. K. Srivastava, A. K. Tiwari and R. K. Shukla, "A Survey: Approaches to Facial Detection and Recognition with Machine Learning Techniques," *Proceedings of Second Doctoral Symposium on Computational Intelligence,* p. 902, 2021.

[5] A. Najibi, "Racial Discrimination in Face Recognition Technology," *SITN,* p. 1, 2020.

[6] R. Belaroussi, P. Foucher, B. Soheilia, P. Charbonnier and N. Paparoditis, "Road Sign Detection in Images: A Case Study," *International Conference on Pattern Recognition,* p. 5, 2010.

[7] S. A. Sanchez, H. J. Romero and A. D. Morales, "A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework," *IOP Conference Series: Materials Science and Engineering*.

[8] Z. Keita, "YOLO Object Detection Explained," datacamp.com, September 2022. [Online]. Available: https://www.datacamp.com/blog/yolo-object-detection-explained.

[9] X. Li, F. Sarro, Z. Chen, Y. Zhang and X. Liu, "Dark-Skin Individuals Are at More Risk on the Street: Unmasking," *Dark-Skin Individuals Are at More Risk on the Street: Unmasking*.

[10] D. C. Everen, "The History of OCR," Veryfi, 28 February 2023. [Online]. Available: veryfi.com/ocr-api-platform/history-of-ocr/. [Accessed 18 September 2023].

[11] S. V. Rice, F. R. Jenkins and T. A. Nartker, "The Fourth Annual Test of OCR Accuracy," Las Vegas, 1995.

[12] L. Quir´os and E. Vidal, "Evaluation of a Region Proposal Architecture for Multi-task Document Layout Analysis," *arXiv,* p. 11, 2021.

[13] D. S. Vijayarani and M. A. Sakila, "Performance Comparison of OCR Tools," *Internation Journal of UbiComp,* vol. 6, no. 3, p. 11, 2015.

[14] Information Science Research Institute, "Annual Research Report," *N/A,* p. 96, 1993.

[15] J. Muller and K. Dietmayer, "Detecting Traffic Lights by Single Shot Detection," *21st International Conference on Intelligent Transportation Systems (ITSC),* p. 8, 2018.