Report #18-UT-066

NSTSCE

National Surface Transportation Safety Center for Excellence

Application of Proximity Sensors to In-vehicle Data Acquisition Systems

Ujwal Krothapalli • Loren Stowe • Zac Doerzaph • Andy Petersen

Submitted: May 2, 2018

Technology Impairment Mobility Infrastructure

Housed at the Virginia Tech Transportation Institute
3500 Transportation Research Plaza • Blacksburg, Virginia 24061

ACKNOWLEDGMENTS

The authors of this report would like to acknowledge the support of the stakeholders of the National Surface Transportation Safety Center for Excellence (NSTSCE): Tom Dingus from the Virginia Tech Transportation Institute, John Capp from General Motors Corporation, Chris Hayes from Travelers Insurance, Martin Walker from the Federal Motor Carrier Safety Administration, and Cathy McGhee from the Virginia Department of Transportation and the Virginia Transportation Research Council.

The NSTSCE stakeholders have jointly funded this research for the purpose of developing and disseminating advanced transportation safety techniques and innovations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

DPM	deformable parts-based model
HOG	histogram of oriented gradients
GPU	graphics processing unit
PCP	percentage of correct parts
RBF	radial bias function
RGB	red, green, blue
SHRP 2	Second Strategic Highway Research Program
SVM	support vector machine
VGA	video graphics array
VTTI	Virginia Tech Transportation Institute
VTTIMLP	Virginia Tech Transportation Institute machine learning pose

CHAPTER 1. INTRODUCTION

Using computer vision to recognize human activity is a well-studied application area. When applied to drivers, activity recognition can provide researchers with vital information about driver distraction, which is a serious problem affecting the safety of drivers and other roadway users. Distracted driving involves deviation from standard driving and can include actions such as moving the eyes away from the road and using the forelimbs to perform various tasks (Dingus & Klauer, 2008). Detecting such actions can help researchers identify causal factors that contribute to safety-critical events such as a crashes or near-crashes. Currently, human data reductionists manually review data and add annotations to describe such driver behaviors alongside the captured data streams. Accomplishing this task with algorithms that can mimic the data reduction process will reduce human labor and speed up the process significantly. This report will describe a research project which developed and applied pose estimation methods to perform activity recognition within the vehicle context. Included is a discussion of the challenges encountered in pose estimation and how careful review of the challenges led to an alternative method of performing driver activity recognition.

Machine learning is the science of understanding and detecting patterns that exist in data. These techniques can be applied to a broad range of data to capture various patterns that may be imperceptible to the human eye and are also useful when the automation of tasks needs to be similar to the human decision-making process. Classic computer vision methods of pattern detection can be augmented with machine-learning methods to make them more robust. This report will describe pose estimation algorithms based on a combination of machine vision and machine learning, including the workflow and iterative process that led to the final driver activity recognition solution. The machine-learning algorithms used in this report are called classification algorithms; they help in the building models that simultaneously maximize inter-class distance and minimize intra-class distance. Feature extraction methods also help minimize the intra-class distance by using a modified representation of the original input. Figure 1 shows a brief overview of the project's workflow.

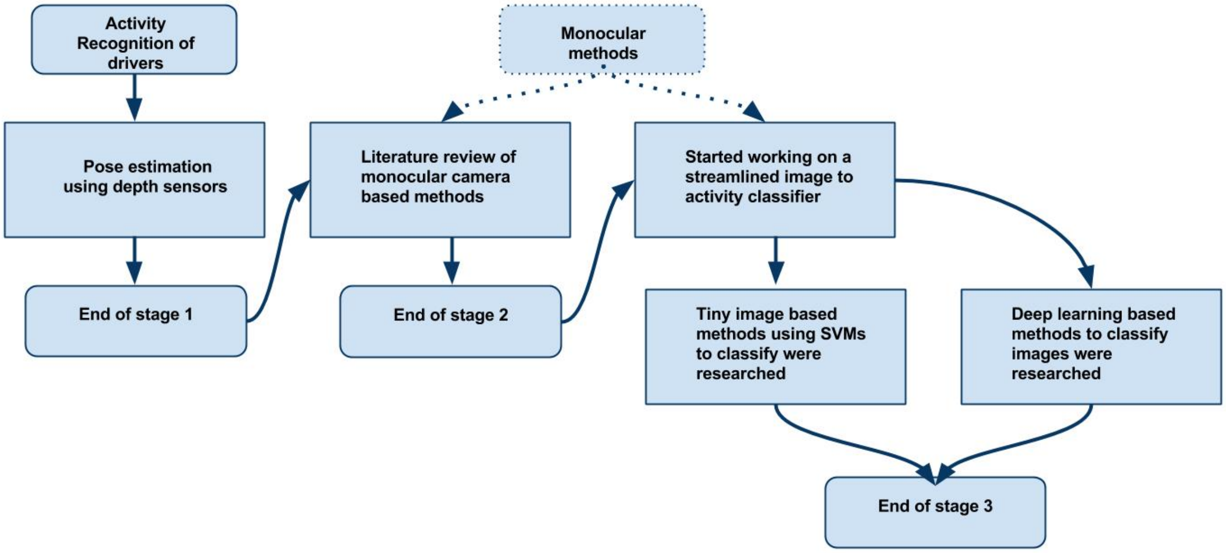
Activity Recognition of drivers
Monocular methods
Pose estimation using depth sensors
Literature review of monocular camera based methods
Started working on a streamlined image to activity classifier
End of stage 1
End of stage 2
Tiny image based methods using SVMs to classify were researched
Deep learning based methods to classify images were researched
End of stage 3

Figure 1. Flowchart. Brief overview of the project's workflow.

The workflow shown in Figure 1 provides an overview of the work performed during this project. A broad spectrum of tools was applied and developed to accomplish the project goals of being able to automatically detect driver activity in an efficient manner using a wide variety of sensors. The researchers started with depth-based (3D) solutions in Stage 1 and arrived at a monocular (2D) solution, which became a more commonly used approach during the course of this effort. A state-of-the-art method known as deep learning was also explored toward the end of the project, including a proof of concept. Chapter 3 of this report will go into these methods in greater detail.

CHAPTER 2. BACKGROUND AND LITERATURE REVIEW

The pose estimation of an articulated object (i.e., a human) is a difficult task; such a model must cope with changes in lighting, shadows, clothing, etc. A fully functional pose estimation tool can automate the task of monitoring people performing manual activities (e.g., workers operating machinery or drivers piloting a vehicle). One of the more popular ways to solve the problem of pose estimation is by applying individual templates representing different parts of the body and placing geometric constraints on the pairs of templates corresponding to different body parts. (Felzenszwalb & Huttenlocher, 2005). The dominant methods for human pose estimation describe the locations (in pixels) and orientations (with respect to the plane of the image) of the body parts by a set of parameters. Approaches based on parts that are parameterized purely based on their locations have also returned good object detection results (Bourdev & Malik, 2009; Felzenszwalb, Girshick, McAllester, & Ramanan, 2010).

The next few sections will describe a broad range of topics that have been explored by computer vision researchers to solve the problem of pose estimation. These sections will attempt to provide in-depth background information about the elements involved in computer vision and recognizing human activity. Each section is also intended to model the problem in various ways to help explain how we improved the accuracy and efficiency of the methods involved in pose estimation.

ARTICULATED POSE ESTIMATION

Estimating people’s poses is a difficult computer vision problem to solve, especially given the variation in the size and shape of the hands and forelimbs among different people. Additional variance is also introduced by clothing, camera viewpoint changes, and foreshortening (the effect of objects appearing smaller due to their angle relative to the viewer) of individuals’ body parts. To capture all of these variations requires building a complex machine-learning model, which in turn requires a larger dataset from which to learn all the parameters. The general rule of thumb in machine learning is that the size of a training dataset should be at least five times the number of parameters to be learned. Thus, a complex model with many parameters will require more training data than a less complex model. The approximation used by Yang and Ramanan (2013) models the variation caused by foreshortening by approximating the effects of foreshortening using a mixture of pictorial structures that do not encode orientation. The authors use a dynamic programming approach to share the computation across mixtures at the time of inference. By capturing the effects of global geometry on the local parts and partial occlusion of the parts, their model improved on previously established baselines.

Another approach to pose estimation, taken by Shotton et al. (2013), is to use a combination of a depth-based sensor and random forest classifier to make a prediction. These authors employed a random forest classifier in their study to identify the body parts in humans. Random forests are a collection of decision trees. When an input is presented at the top of the tree, it is propagated to the next node in the hierarchy of a tree-like structure, ultimately classifying the input into one of the classes at the bottom of the tree. Such an approach would work in real-time to detect different body parts, thereby solving the problem of detection and localization. Depth sensors like the Microsoft Kinect provide 3D information along with red, green, and blue (RGB) information.

These types of depth information can be used for pose estimation, and the 3D nature of the data allows for easy clustering of the points on the objects compared to monocular methods.

PART TEMPLATES

Part-based models have been used for rigid body recognition. These approaches model the distance between various parts of an object as spring displacements as shown in Figure 2, and as a result can accommodate changes in viewpoint to a certain degree. As there is no movement among the different parts of the object, a global mixture model can capture the variation in different views (Felzenszwalb et al., 2010). The approach taken by Yang and Ramanan (2013) is suitable for non-rigid object detection; using mixtures (a collection of parts) for the local part templates, they generate an exponential number of global mixtures. A prior—defined as the probability distribution over a set of distributions which expresses a belief in the probability that a particular distribution is the distribution generating the data—for co-occurrence of these local parts is computed to accommodate for the likeliness of a local part co-occurring with another part (usually the neighboring part). This approach then employs supervised learning to learn the structured model. The structure or grammar (elbow is connected to the forearm, which is connected to the wrist, etc.) of the model is captured during the learning process.

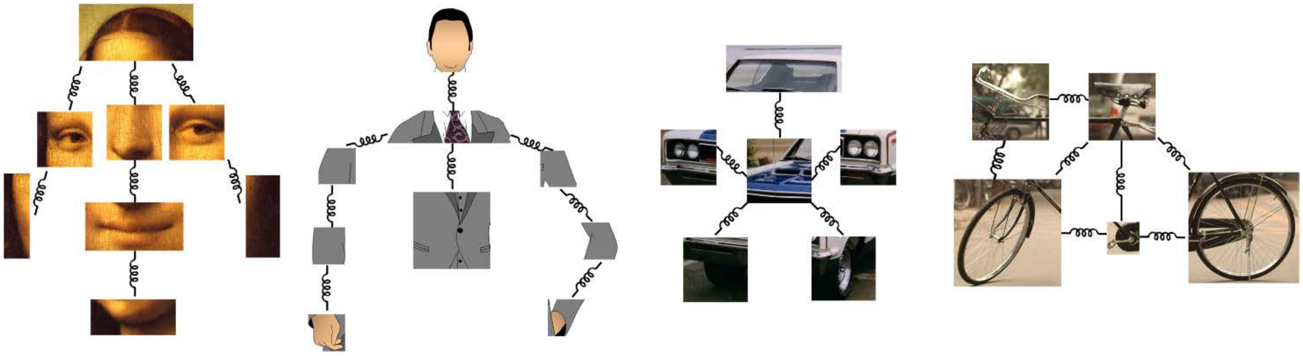

Figure 2. Image. Pictorial structure of various object models with deformations modeled as springs.

ACTIVITY RECOGNITION

Recognizing the human actions present in a monocular image is a simple task for humans, as we possess the innate ability to apply reason to the scene dynamics (actions are usually dynamic). Human activity recognition using computer vision and machine-learning techniques is more difficult, and much work has been done in this area to improve activity recognition methods (Everts, van Gemert, & Gevers, 2013; Raptis & Sigal, 2013; Tian, Sukthankar, & Shah, 2013). The implementation of Everts et al.'s (2013) technique is based on color space modifications to obtain spatiotemporal interest points. Work done in Raptis and Sigal (2013) and Tian et al., (2013) depends on spatiotemporal information from a sequence of frames, as estimating a person's pose in a sequence of frames is less difficult than estimating their pose in static images. Another method involves spatiotemporal reasoning and tracking used in a video sequence to build a model, as in work done by O'Rourke and Badler (1980) and Rohr (1994), but results suffer when severe lighting changes occur.

All of the methods described above have deficiencies that are complementary to each other. For example, an algorithm that uses spatiotemporal information would be unable to cope with sudden changes in lighting, whereas a part-based model might be able to deal with lighting changes without trouble. A spatiotemporal model is very efficient at tracking objects, but a simple part-based approach has no notion of tracking and hence would have to be very efficient, as it relies on detection at time step to perform the same task of activity classification.

Variations of the methods discussed above were tested in a controlled environment to understand their strengths and weaknesses. The following sections will provide more details about the methods that were tested. Thorough testing and evaluation of these methods led researchers to take a final human activity recognition approach that is in contrast to these methods. Only a single frame was used at test time to classify the activity occurring in the image.

EFFICIENCY AND EFFICACY

In addition to accuracy, an important feature of the activity detection algorithm is to be as efficient as possible so that it can be used to make real-time predictions. The C++ implementation of Yang and Ramanan’s (2013) pose estimation can predict the pose of a human’s upper body in a given image in less than half a second. In this pyramid-based search method, a search for all the body parts results in a robust detection framework. A large number of possible poses are evaluated and the highest scoring pose is picked. Further improvements in speed can be obtained by using a graphics processing unit (GPU)–based feature computation and using a cascade-of-rejectors-based framework.

One of the standard ways of evaluating the efficacy of pose estimation algorithms is by computing the percentage of correct parts (PCP), a method first implemented by Ferrari, Marin-Jimenez, and Zisserman (2008). Numerous other implementations of this method also exist because of the vague specifications provided by the authors, as well as the fact that the implementation assumed that people were detected in the first place. Other implementations include a more robust evaluation method developed by Yang and Ramanan (2013), and a system used by Shotton et al. (2013), which can run at over 200 frames per second (fps) on customized GPU hardware.

The final solution arrived at for this report takes a rather computationally cheap yet effective approach to solving the problem of classifying a driver’s actions inside a vehicle. By using tiny image (a downsampled version of the image) features and a multi-class classifier the researchers were able to classify different actions with a relatively high degree of accuracy.

SPATIAL STRUCTURE

Markov random fields and graph-based methods can be used to solve a tree-based method. A tree-based method can be used to build a spatial structure. Tree-based methods commonly suffer from a double counting problem where the same limb gets detected twice, completely neglecting the other limb in the image (Felzenszwalb & Huttenlocher, 2005). Using loopy belief propagation to model the spatial structure can minimize these problems (Sigal & Black, 2006). Instead of terminating the tree at the last branch, the branches are connected to their symmetric counterparts (left to right), thereby eliminating the double counting problem. Yang and Ramanan (2013) use a tree structure to encode the spatial structure and train their model discriminatively using positive and negative sets of images.

LEARNING

Most of the contemporary work related to learning in pose estimation has involved training the local parts independently. Several authors have used independent parts in a boosted detector framework (Andriluka, Roth, & Schiele, 2009; Sapp, Toshev, & Taskar, 2010; Singh, Nevatia, & Huang, 2010). Yang and Ramanan (2013), however, rely on a joint learning framework to improve the accuracy of detection. Intuitively, this makes sense, as detecting only one weak template at a time will result in a poor performance compared to summing up the scores for multiple templates.

FEATURE EXTRACTION

Computer vision algorithms rely on a specific feature extraction method depending on the application. The most common feature extraction methods for pose estimation are edges (Mori & Malik, 2002; Sullivan & Carlsson, 2002), background/foreground models (Ferrari et al., 2008), and superpixels. Dalal and Triggs (2005) used histograms of oriented gradients (HOGs), first employing them in a human/pedestrian detection framework. HOGs bin the orientations in a given cell to provide a descriptor for creating a library of templates. They compute quickly and are generally invariant to minor changes in lighting conditions. Figure 3 shows the HOG filters used by Yang and Ramanan (2013) as part templates. Each of these features has its own modes of failure and success; therefore, combining them will return better results. The Methods section of this report will explain in greater detail the importance of selecting complementary features.

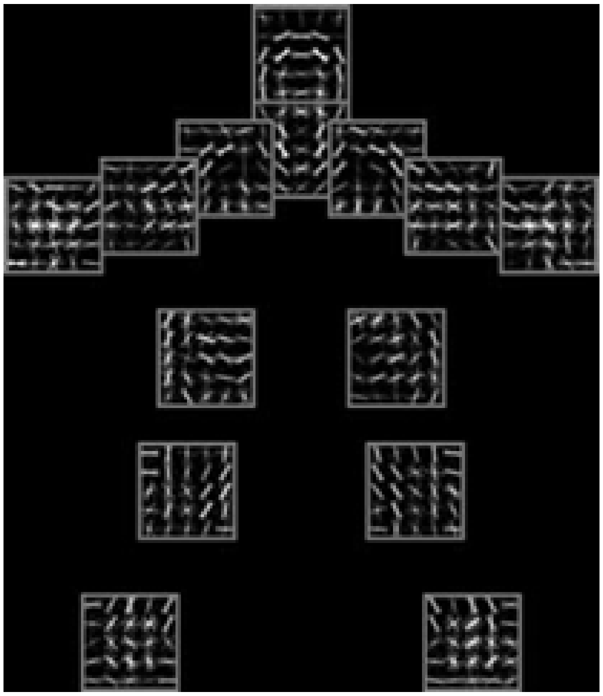

Figure 3. Image. HOG templates visualized for a 14-part human model from Yang and Ramanan's (2013) implementation.

TEMPLATE SIZE

On a global scale, a denser computation might be required to capture most of the information necessary for an object detection framework (e.g., instead of choosing an 8 × 8 pixel cell for the HOG computation, a 4 × 4 pixel cell is used, thereby increasing the resolution of the HOG representation). However, when local parts are used, dense computation can be avoided by using multiple small-sized parts. Poselets are another class of part filters that employ very specific image patches to build a library of filters. Large-scale parts can be used effectively in a coarse-to-fine detection framework (Sun & Savarese, 2011; Wang, Tran, Liao, & Forsyth, 2012). Small local templates with joint learning are effective against clutter that might look like a limb when put out of context. Yang and Ramanan (2013) follow the traditional pictorial structure model and capture the variance using a mixture of parts (Felzenszwalb & Huttenlocher, 2005; Ullman, 2009). A pose is obtained by performing an AND operation across all local parts and performing an OR operation across all the corresponding mixtures.

CHAPTER 3. METHODS

This section will explain the workflow that led to the final driver pose estimation algorithm. Figure 4 shows a detailed project flowchart. After an initial literature review, discussed above, the researchers focused efforts on performing thorough testing of various methods and evaluated the pros and cons of each at every step. A variety of datasets were collected as a part of this study and data reduction was performed in multiple ways to accommodate different algorithms. The researchers deemed the three following methods to be the most viable and spent significant time and effort refining them: (1) depth-sensor-based pose estimation; (2) deformable parts-based model (DPM); (3) tiny-image-based driver activity classifier. Other methods were found to be less productive and their deficiencies are discussed in detail herein. The first two methods were used to generate the joint locations, and then classify the points into various classes. The third method is a more streamlined version that predicts the driver's activity from a single image. Of all the methods that the researchers employed, the deep-learning-based solution performed the best, though it required specialized hardware to train on. The "shallow" counterpart also performed this task well without reliance on specialized hardware.

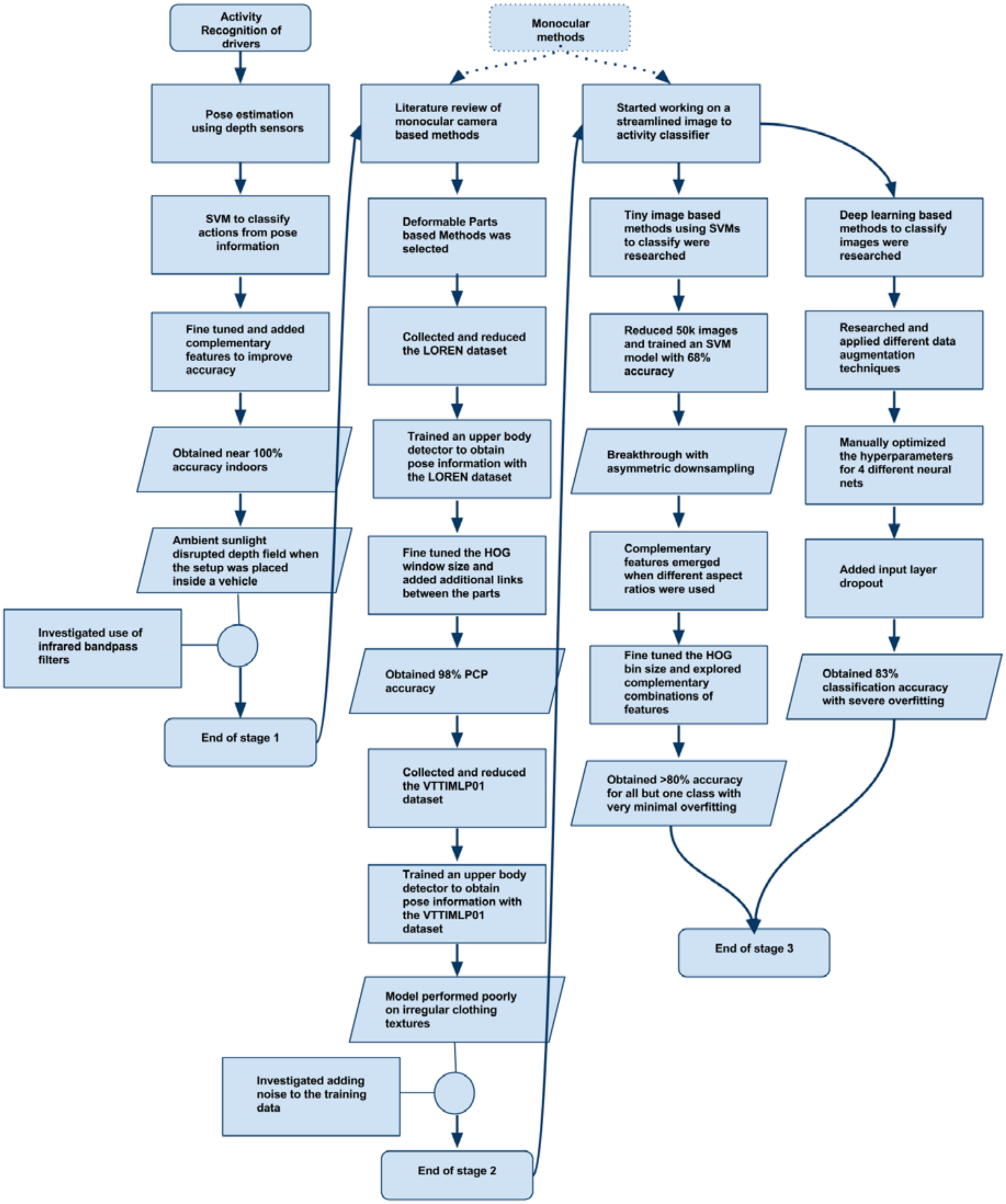
Activity Recognition of drivers
Pose estimation using depth sensors
SVM to classify actions from pose information
Fine tuned and added complementary features to improve accuracy
Obtained near 100% accuracy indoors
Ambient sunlight disrupted depth field when the setup was placed inside a vehicle
Investigated use of infrared bandpass filters
End of stage 1
Monocular methods
Literature review of monocular camera based methods
Deformable Parts based Methods was selected
Collected and reduced the LOREN dataset
Trained an upper body detector to obtain pose information with the LOREN dataset
Fine tuned the HOG window size and added additional links between the parts
Obtained 98% PCP accuracy
Collected and reduced the VTTIMLP01 dataset
Trained an upper body detector to obtain pose information with the VTTIMLP01 dataset
Model performed poorly on irregular clothing textures
Investigated adding noise to the training data
End of stage 2
Started working on a streamlined image to activity classifier
Tiny image based methods using SVMs to classify were researched
Reduced 50k images and trained an SVM model with 68% accuracy
Breakthrough with asymmetric downsampling
Complementary features emerged when different aspect ratios were used
Fine tuned the HOG bin size and explored complementary combinations of features
Obtained >80% accuracy for all but one class with very minimal overfitting
Deep learning based methods to classify images were researched
Researched and applied different data augmentation techniques
Manually optimized the hyperparameters for 4 different neural nets
Added input layer dropout
Obtained 83% classification accuracy with severe overfitting
End of stage 3

Figure 4. Flowchart. Workflow process for the duration of the project.

DEPTH-SENSOR-BASED POSE ESTIMATION

This section explains Stage 1 of the pipeline described in the workflow illustrated by Figure 4.

One of the first methods we investigated was the use of a depth sensor made by PrimeSense to determine the location of the driver's joints. After performing extensive indoor testing and developing a prototype application, the setup was placed inside a vehicle.

The depth sensor used proprietary software and a pre-trained model to detect humans and append a "skeleton" on top of the detected human body. One of the depth sensor's key technologies was structured lighting, which used an infrared laser at ~910 nm wavelength. The sensor projected a dynamic pattern with binary stripes to compute the depth of the scene as shown in Figure 5. A pre-trained, random-forest-based classifier was used to detect humans and detect various joint locations. Figure 6 shows an example of the depth sensor's tracking and detection, which is also explained in detail by Shotton et al. (2013).

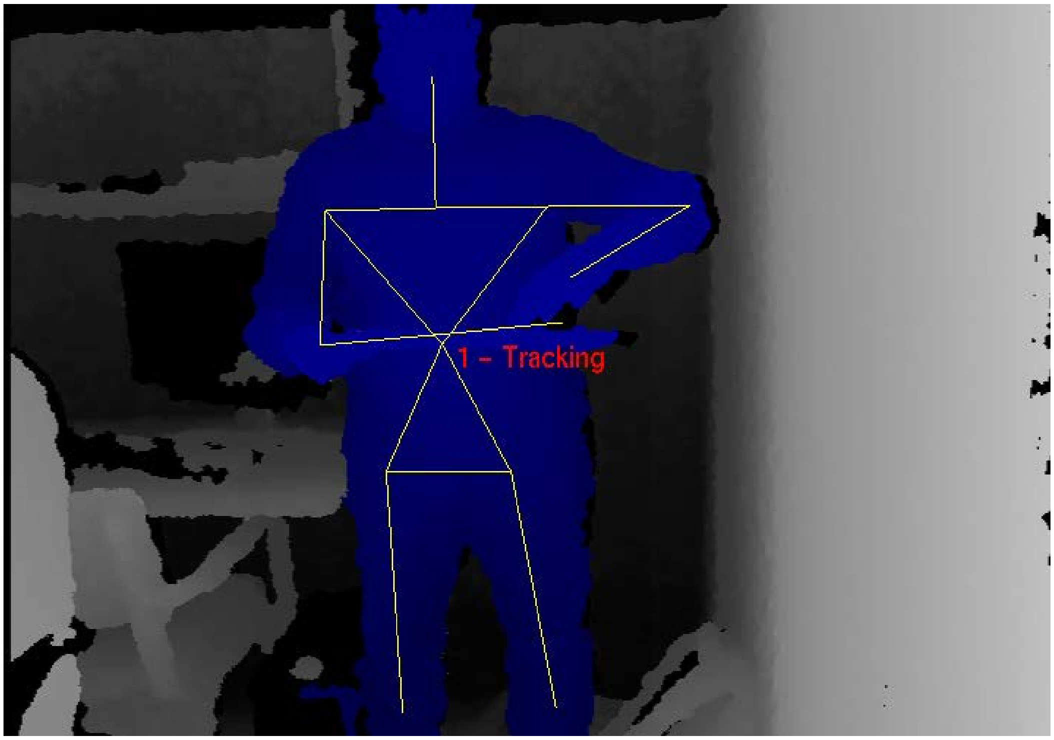

Figure 5. Image. Output of the depth sensor in an indoor environment.

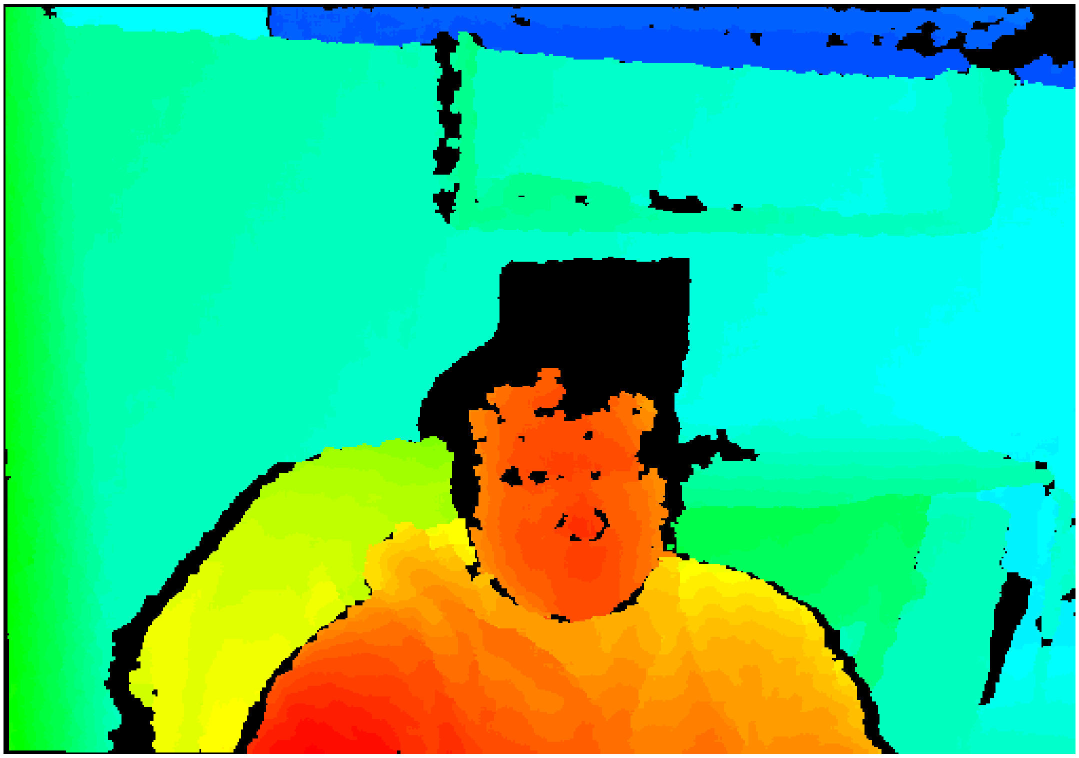

Figure 6. Image. Depth map computed by the depth sensor in an indoor environment.

After obtaining the joint locations, a support vector machine (SVM)–based classifier was trained with features from the 3D points for various classes (each class comprised a specific action). This work was extended to detect various poses indoors, and the authors were able to classify about 30 different poses with a near 100% accuracy (Krothapalli & Christie, 2013).

However, once the system was placed inside a vehicle in daylight conditions, the skeleton detection consistently failed and the depth map had severe artifacting. A further analysis identified the cause to be ambient sunlight. The sun's infrared spectrum is a wide bandwidth signal, and solar radiation is dominant at almost all of the infrared wavelengths. The idea of a band pass filter was briefly considered, but it was determined that the presence of ambient infrared rays, which are about three to four times more powerful than the output of the sensor's laser, would still affect the quality of the depth map, and a more robust and simple architecture that was relatively impervious to ambient sunlight was needed to solve the problem.

DEFORMABLE PARTS-BASED MODEL (DPM)

Monocular cameras were identified as a possible alternative pose detection solution. Computer vision using monocular cameras has made significant advances in the past few decades. As a result, monocular cameras are used more widely today. The cost of these cameras is also relatively low compared to depth sensors. In addition, most monocular cameras can compensate for extremely bright objects (like the sun) to a certain extent. These factors all contributed to a change in research direction, with the end result being the decision to use monocular cameras for activity recognition.

The first step in Stage 2 of the pipeline, therefore, is pose estimation in the 2D domain (similar to the 3D pose estimation step in Stage 1). This section will discuss the work done in Stage 2 of the pipeline as depicted in Figure 4. Yang and Ramanan's (2013) implementation was used to develop a pose estimating algorithm. The actual implementation had 10 points across the upper body and was trained on the Buffy Stickmen and PARSE datasets (Eichner, Ferrari, & Zurich, 2009).

Datasets

The researchers decided to collect real-world data for this project; this decision was based on the experience that data collected under controlled conditions may not capture environmental factors that affect the quality of data in the real world. The resulting LOREN dataset consists of about 200 images that were reduced to have 10 different points marked on the driver's body (head, forelimbs, and torso) as part of a sample dataset. The reduction is shown in Figure 7. To collect these images, a participant drove around the parking lot in daylight, and images were collected using a video graphics array (VGA)–resolution USB camera. The angle of the camera was maintained such that the forelimbs of the participant driver were visible in all the frames. The same reduction was employed for the Buffy Stickmen dataset, which has been widely used for building upper body detectors by the computer vision community.

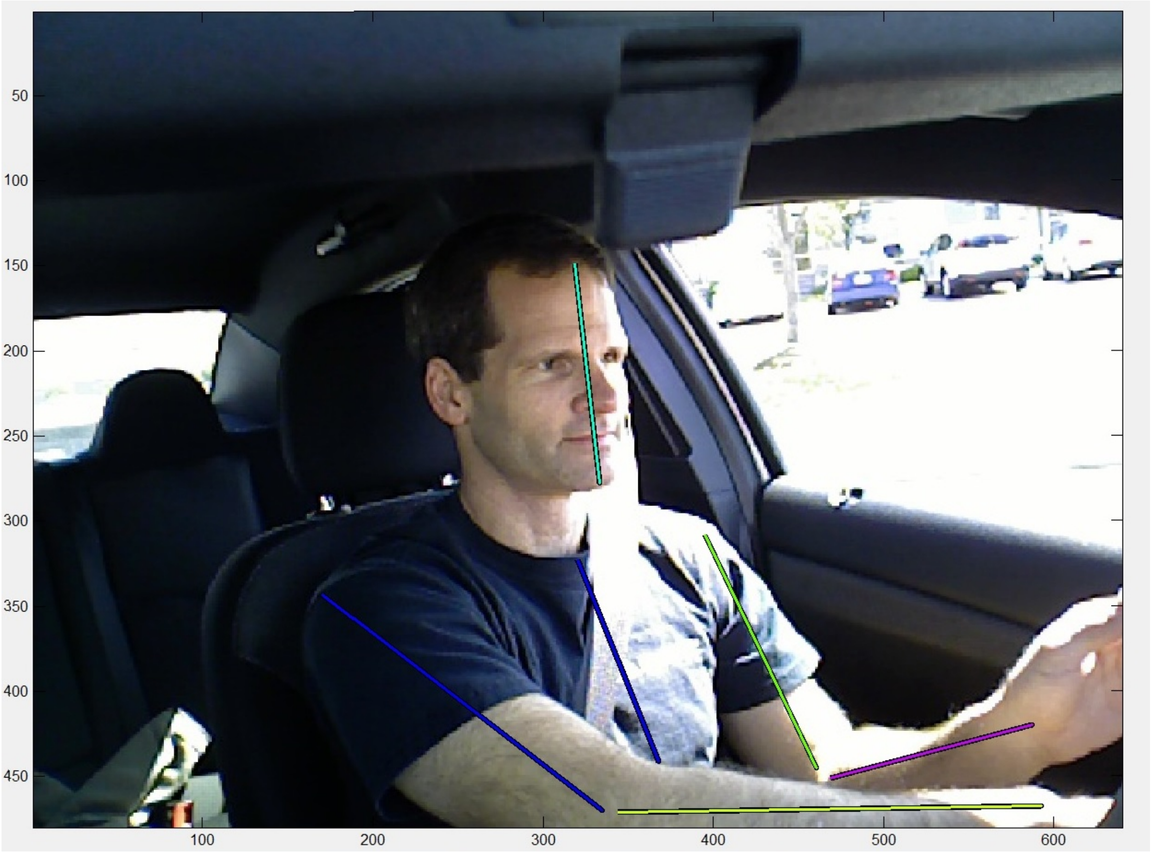

Figure 7. Image. Reduction of the joint locations overlaid on the driver in the LOREN dataset.

The pre-trained model from the Buffy dataset performed poorly on the LOREN dataset. These models suffer from a limitation known as “dataset bias,” which arises because of the variation in the collection methods, instrumentation used, and other factors that are characteristic to the dataset. The solution is to retrain the model with new data. In some cases, the new data can be added to the existing data instead of replacing it before retraining. After retraining with the LOREN dataset, the torso points were removed, as the model became confused between the limbs and the seatbelt. To further improve accuracy, additional midpoints capturing the co-occurring rigid shapes inside the vehicle were needed. The solution was to add additional linkages as described in Figure 8.

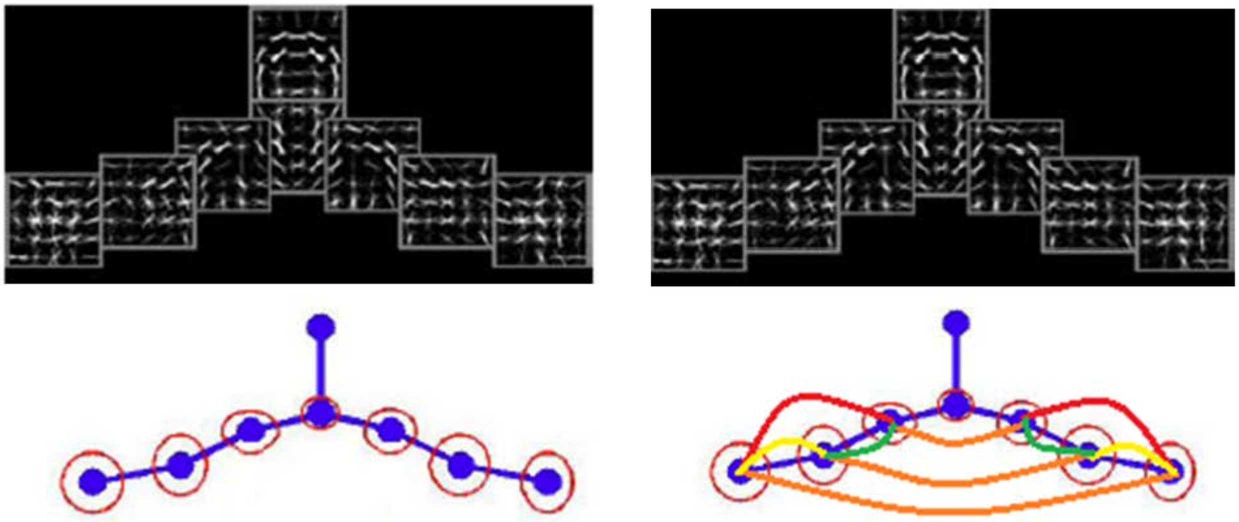

Figure 8. Images. Top left – part filters for each human part; bottom left – tree structure capturing the kinematic chain; bottom right – kinematic chain used by Yang & Ramanan (2013); top right – part filters remain the same, but new connections between them will improve accuracy.

The DPM works by building a kinematic chain or tree that captures various parts of the human body. This work is directly related to the part-based methods discussed in Chapter 2. The “grammar” (kinematic chain) is explicitly provided to the model and can be changed based on the application. Each body part is represented by a collection of HOG filters/templates learned during training. The following sections describe, in detail, the workings of the DPM used by Yang and Ramanan (2013).

The Objective Function

To capture the various deformations and viewpoint changes, an approximation can be made by using a mixture model. For an image *I*, let the pixel location of part *i* and the corresponding mixture component (t_i) be $l_i=(x,y)$. The mixture component is essentially a rotated version of the template. For a *K* number of mixtures, we have $t=t_1,t_2....t_k$. The scoring function would also have to capture the co-occurrence relationship for all the parts.

$$S(t)=\sum_{i\in V} b_i^{t_i}+\sum_{ij\in E} b_{ij}^{t_i,t_j} \tag{1}$$

where $b_i^{t_i}$ is a prior for the mixture model for part i and the co-occurrence is captured by $b_{ij}^{t_i,t_j}$. The value would be high for mixtures that are consistent and low for any inconsistent orientations. Rigidity between the rigid parts of the body is also encoded through this co-occurrence computation. To compute the score for various local part templates and location, the following equation is used.

$$S(I,l,t)=S(t)+\sum_{i\in V} w_i^{t_i}.\varphi(I,l_i)+\sum_{ij\in E} w_{ij}^{t_i,t_j}.\psi(l_i-l_j) \tag{2}$$

The HOG feature for location l_i in image I is defined as $\varphi(I,l_i)$. To capture the deformation among the parts, $\psi(l_i-l_j)=[dxdx^2dydy^2]^T$ is defined, where $dx=xi-xj$ and $dy=yi-yj$ are the difference in x and y location of parts. The appearance model is encoded by computing a score for placing a template and the corresponding mixture model at a location. The second summation is the major improvement that Yang and Ramanan (2013) introduced. This spring model limits the placement of the parts by using a different deformation constraint based on the mixture type. The $w_{ij}^{t_i,t_j}$ term captures the rest location and the rigidity of the parts. By using different spring constraints for different mixtures, the resulting pose is more accurate. The implementation of Yang and Ramanan (2013) uses a constrained set of springs by introducing a simplification, $w_{ij}^{t_i,t_j}=w_{ij}^{t_i}$. This simplification makes the location of a part dependent on the mixture model of the part instead of the parent part.

Inference

This is a maximization problem where $S(I,l,t)$ has to be maximized over locations and mixtures. Because a tree model has been used, the computation can be carried out efficiently by using dynamic programming. Let $z_i=(l_i,t_i)$

$$S(I,z)=S(t)+\sum_{i\in V} \varphi_i(I,z_i)+\sum_{ij\in E} \psi_{ij}(z_i-j_j) \tag{3}$$

$$\varphi_i(I,z_i)=w_i^{t_i}.\varphi(I,l_i)+b_i^{t_i} \tag{4}$$

$$\psi_{ij}(z_i-j_j)=w_{ij}^{t_i,t_j}.\psi(l_i-l_j)+b_{ij}^{t_i,t_j} \tag{5}$$

This model resembles a Markov random field. The maximum can be computed using dynamic programming. This approach will result in multiple detections in a single image. By using a non-maximal suppression strategy, the best detection can be obtained.

Learning

A supervised learning framework was employed by the Yang and Ramanan (2013). Using a set of positive examples and a set of negative examples, a structured prediction model was built. Let $z_n=(l_n,t_n)$ and the scoring function is linear $\beta=(w,b)$. The scoring function can be written as $S(I,z)=\beta.\Phi(I,z)$. Then the model to be learned is given by,

$$\min_{w,\xi_n>=0} \quad \frac{1}{2}\beta.\beta+C\sum_n \xi_n \tag{6}$$

$$\textit{for positive samples} \quad \beta.\Phi(I,z)\geq 1-\xi_n \tag{7}$$

$$\textit{for negative samples} \quad \beta.\Phi(I,z)\leq -1+\xi_n \tag{8}$$

Error! Bookmark not defined.

The above formulation results in all the positive samples scoring more than 1 and all negative samples scoring below −1. The violations on either side of the hyperplane are penalized by the slack parameter, ξ_n. Because the above model is already able to detect between positive and negative samples, the detection problem is solved simultaneously with the pose estimation problem.

Optimization

The formulation given above is a quadratic optimization problem, and the number of constraints are exponential. By using an SVM-based formulation, the only constraints that are also the support vectors will be sufficient to find a solution. These specific types of SVMs are called structured SVMs. Some of the more popular ways to solve such formulations are by using SVMStruct (Tsochantaridis, Hofmann, Joachims, & Altun, 2004) or a stochastic gradient descent approach. Yang and Ramanan (2013) implemented a dual coordinate-descent algorithm to solve the above formulation. This approach finds the support vectors necessary for inference in just a single pass. Lagrangian dual relaxation is employed for efficient computation.

Results

Being able to detect objects and estimate their poses is a challenging problem, especially because of the clutter present in real-world images. Clutter can be a part of the environment (car interior) and/or clothing and accessories (e.g., watches, hats, etc.). The implementation described above was used to determine the driver's upper body pose. The implementation developed by Yang and Ramanan (2013) was used to train an algorithm to predict the poses of participants at the Virginia Tech Transportation Institute (VTTI). About 200 images from the LOREN dataset were annotated to have 10 key points (head top, head bottom, shoulders, elbows, torso top, torso bottom, and wrists). These were used in the positive set; for the negative set, instead of having just the images of backgrounds without people in them, images of empty car interiors were appended. The interiors of cars are shaped to fit to the human form, which caused the HOG templates to get confused on more than one occasion. This problem was eliminated after appending the images of empty car interiors to the negative dataset. The prediction model was then used on 20 test images and PCP scores were computed. The HOG window size played a major role in computing the accuracy; this is related to the size of the limbs in the given image. For the images collected, a window of 12 by 12 gave the best results. The next parameter that affected the accuracy was the number of mixtures (a mixture is a collection of small non-oriented parts). More mixtures can explain the fine articulations better and result in better performance. Using a nine-mixture model along with the 12-by-12 window provided a PCP of 98%. This was improved from 85% PCP when the algorithm was implemented without any fine-tuning. Note, however, that the test set was comprised of a small sample size of only 20 images. One of the factors that helped improve the accuracy was including diverse articulated poses (instead of using similar poses) in the training data. The number of parts also played a role in improving the accuracy. In addition to the eight key points, midpoints were also used to build an 18-part model. The final step in improving accuracy was to modify the tree structure by connecting the shoulder to the wrist by means other than going through the elbow bend. Also, the key points on the torso were removed. By capturing the intermediate parts (inside of the car), the pose estimation was much more accurate (~12% more than the actual implementation of Yang & Ramanan [2013]), as evidenced in the qualitative examples shown in Figure 9. Most of the improvement was in the wrist class.

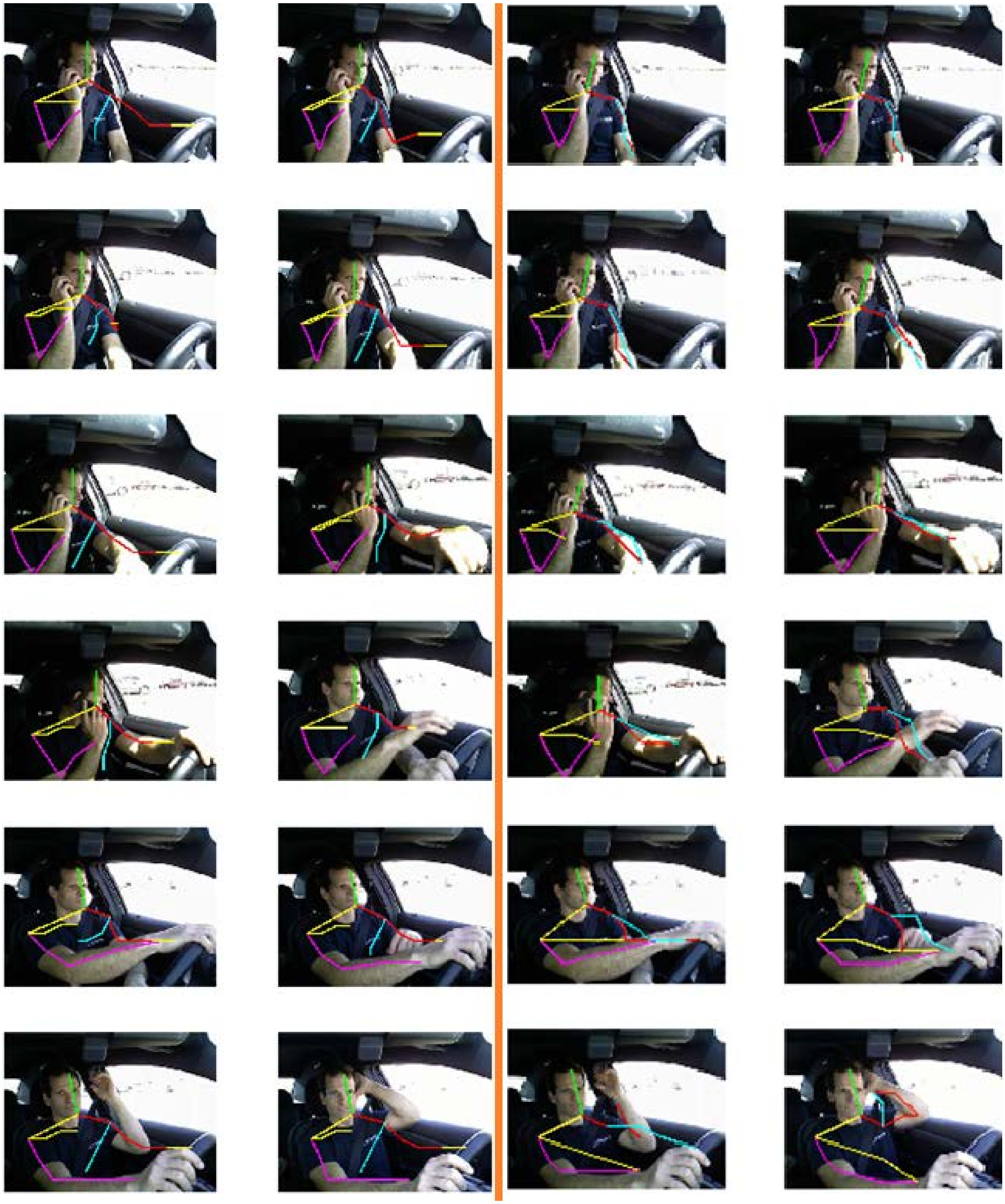

Figure 9. Images. Qualitative results for the upper body pose estimation using Yang & Ramanan's (2013) implementation with (left of center line) unoptimized HOG window size and unmodified tree structure and (right of center line) optimized HOG window size and modified tree structure.

VTTIMLP01 Dataset

The VTTIMLP01 (VTTI Machine Learning Pose) dataset is a collection of about 80,000 images from 25 participants (VTTI employees were used for this data collection effort) in naturalistic driving and simulated naturalistic driving conditions. The authors chose the participants such that there would be diversity in the body shapes across the sample population. This dataset was collected over a month under various ambient lighting conditions. Some of the drivers performed secondary tasks while driving, while all of them performed secondary tasks in a stationary vehicle. The participants were wearing the clothing of their choice. For a more controlled study, more fabric types and textures could be introduced by having participants change their clothing during the study. About 30,000 of these images have eight points (across head and forelimbs) reduced, for which custom tools were developed. A custom camera angle was used in two cars with contrasting interiors such that the forelimbs were visible at all times. A limitation of the algorithm is that the forelimbs have to be visible to achieve a valid pose estimate (making broad application to existing data sets such as the Second Strategic Highway Research Program [SHRP 2] a challenge). See Appendix C for more detailed information on the protocol that was used to collect the VTTIMLP01 dataset. Another 50,000 images were reduced into eight classes, where each class is one of the eight different activities that participants performed while the vehicle was stationary (static) or moving (dynamic). To prevent overfitting and to allow for model complexity, more manual reduction was necessary to allow for the training of a machine-learning model. The amount of training needed depends on the model complexity, and it is important to have sufficient reduction to allow for flexibility in the model space. The following participant activities, performed during data collection, were annotated during data reduction.

1. Eating with the right hand while the left hand is on the steering wheel.
2. Drinking with the right hand while the left hand is on the steering wheel.
3. Talking on the phone with the right hand against the ear while the left hand is on the steering wheel.
4. Adjusting the visor with the right hand while the left hand is on the wheel.
5. Adjusting the center stack controls with the right hand while the left hand is on the wheel.
6. Texting on the phone with the right hand against the ear while the left hand is on the steering wheel.
7. Driving with the right hand on the steering wheel while the left hand is in the participant’s lap.
8. Driving with both hands on the wheel.

Most of the participants were right-handed, so the criteria chosen for reduction, listed above, largely feature tasks performed with the right hand. For the left hand counterparts of the above actions, a different camera angle would better capture the activity. The Eating and Drinking actions were later combined into the same class, as the resolution of the images was too low to distinguish between these two actions, which have extremely similar hand positions.

Table 1. Participants and lighting conditions in the VTTIMLP01 dataset.

Participant ID	Lighting Condition	Collection Format	Sex
1	Daylight	Dynamic	Female
2	Daylight	Dynamic	Male
3	Daylight	Static	Male
4	Daylight	Dynamic	Male
5	Daylight	Static	Female
6	Daylight	Dynamic	Female
7	Daylight	Dynamic	Male
8	Daylight	Dynamic	Male
9	Daylight	Dynamic	Male
10	Night	Static	Female
11	Daylight	Static	Male
12	Daylight	Static	Male
13	Daylight	Static	Male
14	Night	Static	Male
15	Daylight	Static	Male
16	Daylight	Static	Male
17	Daylight	Static	Male
18	Daylight	Static	Male
19	Daylight	Static	Male
20	Daylight	Static	Female
21	Daylight	Dynamic	Female
22	Daylight and Night	Static	Male
23	Daylight	Static	Male
24	Daylight and Night	Static	Male
25	Daylight	Dynamic	Female

Once the VTTIMLP01 dataset was collected, a DPM was trained using 10 drivers and tested on the other 15 drivers. The results suffered, as participants' clothing severely affected the accuracy. Figure 10 shows an example of a case where the model failed to accommodate puffy clothing. This method would work with good accuracy only if the driver's clothing was relatively smooth or if the puffy clothing was present in the training data. To eliminate this problem, the participants in the training data would have to wear clothing of different fabrics, textures, thicknesses, etc. Superimposing noise on top of the training images is another way to simulate a variety of different clothing types and textures.

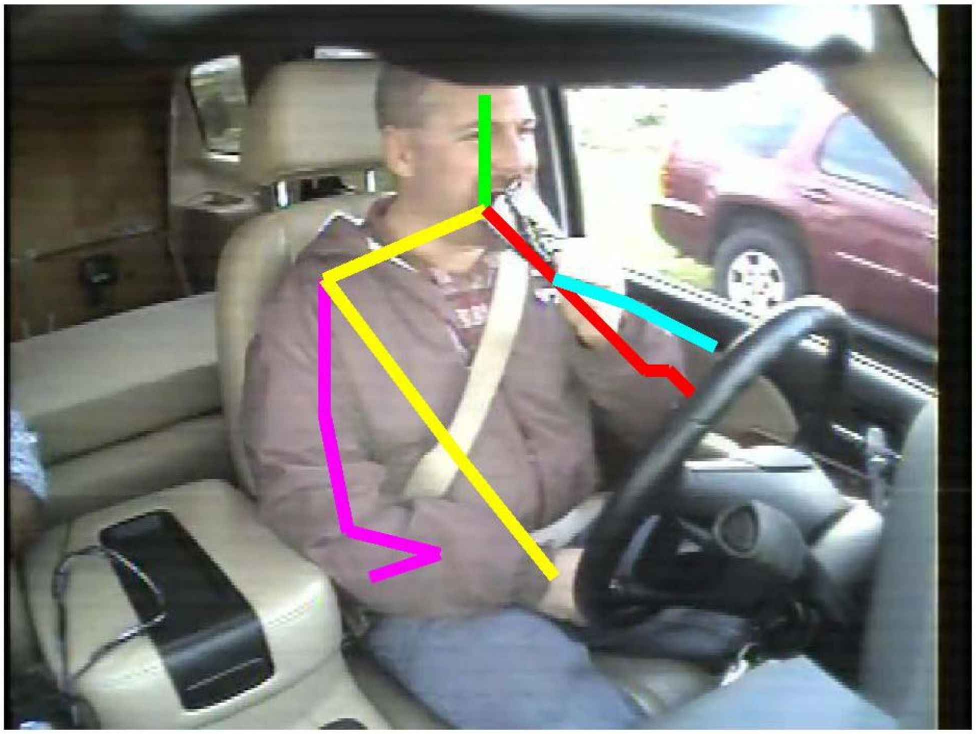

Figure 10. Illustration. An example of the failure case of the pose-estimating algorithm on the VTTIMLP01 dataset.

The researchers decided, at this point, to try a more straightforward approach that was robust not only to changes in lighting, which most spatiotemporal models fail to address, but also to variations in clothing.

TINY-IMAGE-BASED DRIVER ACTIVITY CLASSIFIER

Drawing on researchers' experience with the depth sensors and DPMs, a streamlined approach to perform driver activity recognition was developed. A variety of feature extraction and machine-learning methods, such as gabor filters, color histograms, random forests, and edge detectors, were tested under controlled conditions before being finalized in the workflow. A parallel approach was used to perform activity recognition using both deep and shallow machine-learning methods. The shallow machine-learning methods have a small model space (~10^4 parameters) compared to a deep learning method (~10^8), and going by the rule of thumb (four to five data points for every parameter that needs to be learned), the deep learning models would need relatively large amounts of training data. The deep learning methods, therefore, suffered from overfitting problems and needed more training images. However, the shallow methods did not suffer from this problem. Accordingly, a shallow “tiny-image”-based approach was developed and is being presented herein, along with HOG features, as the final activity detection solution. This method involves very few operations on the image and generalizes well compared to the other methods that were tested.

Feature Extraction

A “tiny image” is the simplest feature extraction that is employed in computer vision—it is essentially a low-resolution version of the given input image. The VGA resolution image was converted to grayscale and a square crop was extracted. Figure 11 shows a distribution of the drivers’ joint locations in the VTTIMLP01 dataset; this information was used to extract a square crop that captured the limbs of all the participants in the given image. Since the camera angle was known, the crop was able to capture the driver at all times. The image was then downsampled with custom criteria (asymmetric downsampling) to maximize the accuracy of class activity. Then the image was contrast normalized. The intuition behind asymmetric downsampling is that some of the features in the image get magnified and others get diminished compared to the image, which maintains its aspect ratio, and this effect can be tuned to improve individual class accuracies. After downsampling, some images tend to have sharp gradient changes, and Gaussian blurring is one of the common techniques for eliminating edges. HOG features were extracted on these images and the resulting multidimensional matrix was then converted to a vector to be passed on as a feature vector. The position of the camera did change between the two vehicles, and the shallow model was able to capture this variation; however, this model is not robust enough to severe changes in the camera position with respect to the driver. The custom downsampling that was employed in this method helped improve the inter-class distance between the seven classes of activity that were present in the VTTIMLP01 dataset.

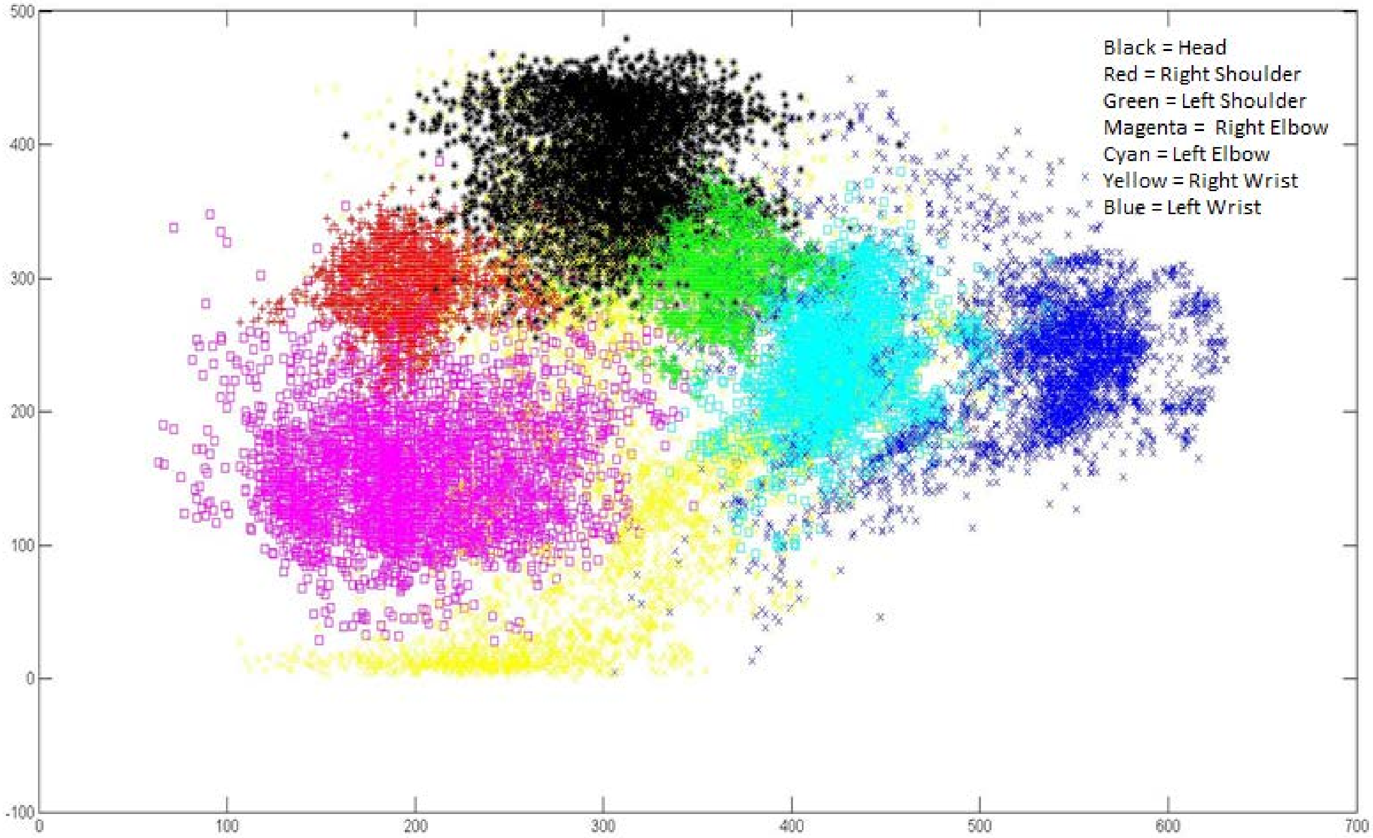

Figure 11. Image. A plot of the various parts of all the drivers in the VTTIMLP01 dataset.

Machine Learning

Feature extraction is the first step in separating the images into activity classes. The modified input data are split into training, validation, and testing datasets. The training data are passed on to a supervised machine-learning algorithm. Supervised means that the activity class labels along with the input are supplied to the algorithm for training and at test time only the input is supplied and the model predicts the class label. The parameters of the machine-learning algorithm are tuned using the validation dataset and the performance assessment is obtained by evaluation of the fine-tuned model on the test dataset. Note that overfitting is a common problem in machine learning; if a machine-learning algorithm performs with very high accuracy on the training dataset but performs very poorly on the validation or test datasets, then the algorithm is most likely overfitting. SVMs, supervised machine-learning models, and associated algorithms were used for this effort. SVMs work by computing maximum margin hyperplanes in the high dimensional space. Figure 12 shows an example of such a process. Chang and Lin's (2011) library is one of the more efficient implementations of SVMs available and it was used for this project. Other implementations are available depending on the programming platform.

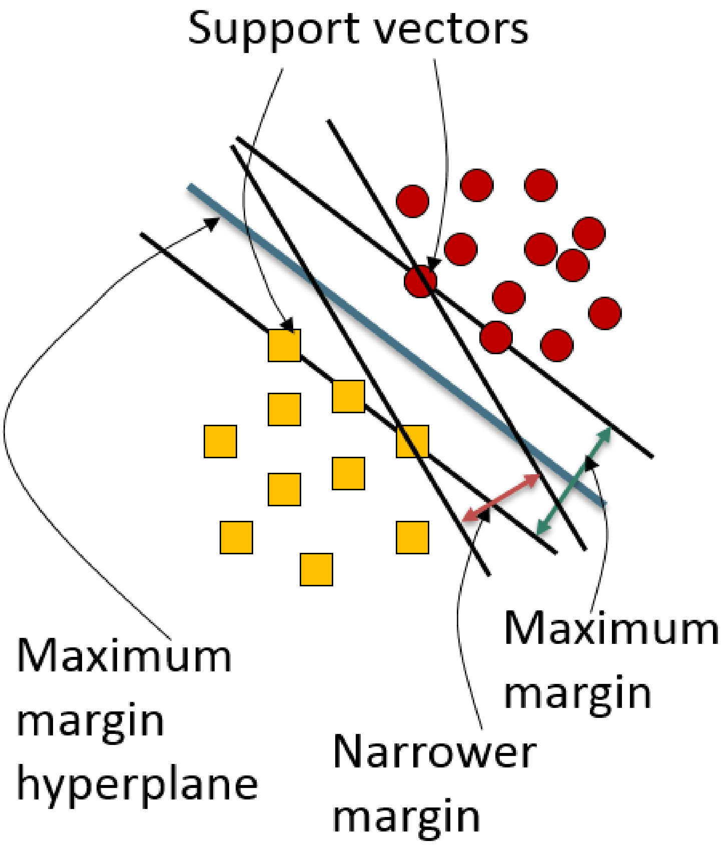

Figure 12. Diagram. Classification with hyperplanes in a classification framework with two classes (orange boxes and maroon dots).

There may be many possible lines separating the two classes, but there exists only one maximum margin hyperplane (equidistant from both classes) that separates the classes. Only a subset of training points defines the maximum margin hyperplane. These points are called support vectors. There may not be a hyperplane in the lower dimensional space that can effectively split the data. In that case, the SVMs can be used to project the data in a high dimensional space using a kernel function, which is commonly known as the "kernel trick." If the classes cannot be separated in the lower dimensional space, they may be separable in one of the higher dimensions. Common examples of kernels include radial basis function (RBF) and hyperbolic tangent (sigmoid). Kernels essentially embed the input data into a higher dimensional space compared to the linear space (if the input is not kernelized) by performing minor mathematical operations.

The flowchart in Figure 13 describes the generic workflow for classifying images using machine-learning algorithms.

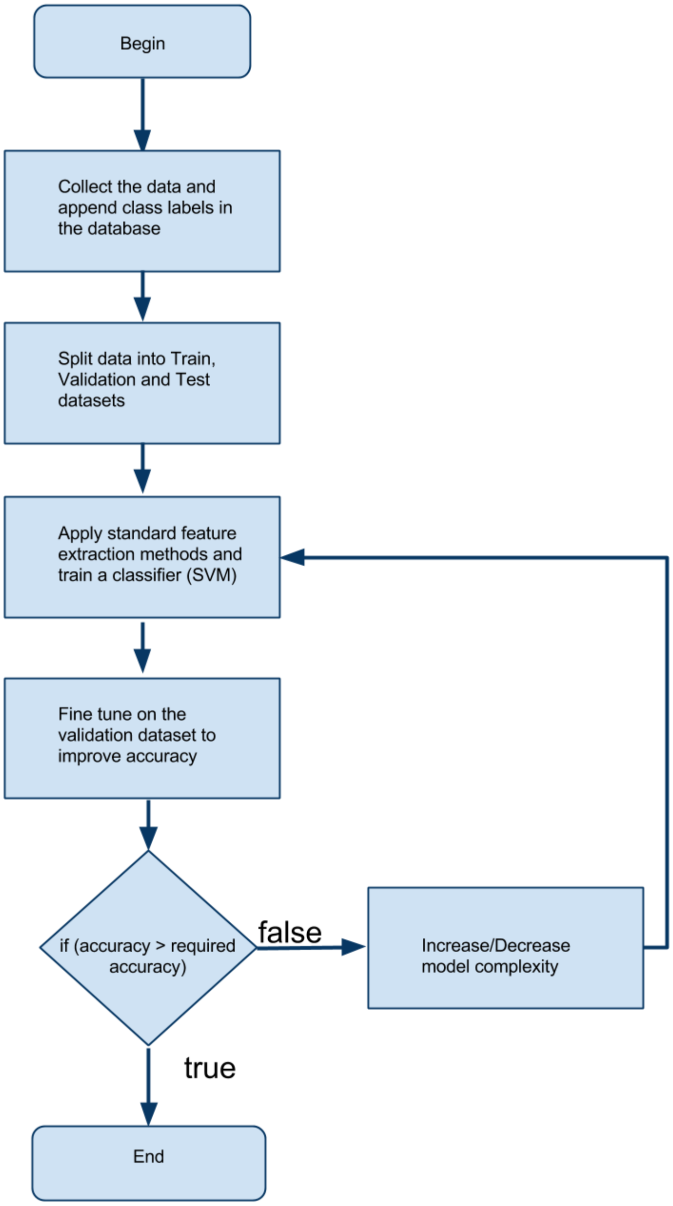
Begin
Collect the data and append class labels in the database
Split data into Train, Validation and Test datasets
Apply standard feature extraction methods and train a classifier (SVM)
Fine tune on the validation dataset to improve accuracy
if (accuracy > required accuracy)
false
Increase/Decrease model complexity
true
End

Figure 13. Flowchart. Training process for a generic machine-learning algorithm for image classification.

Implementation Details

This section will explain the specific process of driver activity recognition using machine learning. The VTTIMLP01 dataset was used for this method. The dataset had seven activity classes (seven different secondary activities being performed by the driver) and 25 participants (including the authors). About 50,000 images, along with the activity class labels (a class label is the name of one of the seven different activities), were used for this method, and joint locations were not necessary for the method used. After applying the feature extraction method mentioned in the Feature Extraction section of this report, the researchers trained an SVM model first without any asymmetric downsampling on the first 20 participants and then tested the model on the last five. The accuracy was close to 70%. Extracting multiple crops based on the centroids of the joint locations across the drivers as shown in Figure 12 did not improve accuracy.

Laplacian pyramids are used in computer vision to capture different spatial frequencies at different levels of downsampling. However, these methods still maintain the aspect ratio of the original image. The researchers experimented with different aspect ratios while downsampling the image in the feature extraction pipeline. There is no literature or prior work in this area. The motivation for employing this method was that when regular downsampling was used, the inter-class distance was too small, affecting the accuracy. As evidenced by the kernel trick, a transformation at image level can help the SVM better classify in the higher dimensional space. By changing the aspect ratio (horizontal stretching) and downsampling asymmetrically, different spatial features were affected. By tuning the aspect ratio, the researchers were able to obtain high class accuracies. The slack parameter of the SVM was fine-tuned to obtain better accuracy, as well as to maintain good generalizability of the model. A radial basis kernel was used in the SVM. The combination of different aspect ratios resulted in a much better performance. It is a common practice in classic computer vision methods to combine complementary ratios to obtain better accuracy; however, better accuracy may not always be achieved. Image classification using human engineered features is completely data driven. The “standard” methods offer a generic pipeline, but the results are not always guaranteed. A proper understanding of the training of the algorithm and controlled experimentation is needed to engineer features.

Note that all the classes were trained based on using a particular hand for a certain activity, as explained in the dataset section of this paper. The model, without any aspect ratio changes to the input image, was able to predict the activity classes with an overall 70% accuracy. To obtain better accuracies for individual activity classes, a separate model was built for each class. This resulted in the model described in Table 2, which shows the overall accuracy of the model to be 74%. The model fails when the forelimbs are not completely visible. The Texting class has the least accuracy of all the models, which is due to the foreshortening effect on the limbs in the given camera angle. For the Eating/Drinking class model, a combination of the features from images sampled at ratios of 3:2, 2:3, and 1:3 were used. The Talking class model was built using features from images sampled at aspect ratios of 4:5 and 5:4. The Visor class model did not use any aspect ratio changes; the image was sampled to 96 pixels by 96 pixels. The Center-Stack class model used an aspect ratio of 2:1. The Texting class model did not use any aspect ratio changes. The One-Hand class model used an aspect ratio of 1:3. The Both-Hands class model used an aspect ratio of 1:4.

Once the features were collected, SVM models were trained and parameters optimized as described above. The fine-tuning of the slack parameter was geared towards minimizing the false positive rate without overfitting. Fine-tuning machine-learning algorithms usually involves minimizing the false positive rate or the false negative rate. In the case of SVMs, this trade-off can be better managed by a technique called hard-negative mining, which works by looking for random patches that cause misclassification and adding them back into the training samples. The Misc. 1 and Misc. 2 classes were introduced to classify the ambiguous images and help minimize the false positive rate.

The following sections describe the results that the researchers obtained on the VTTIMLP01 dataset using the “tiny-image-based driver activity classifier” model. All the tables in this section are confusion matrices, which have the ground truth classes on the vertical axes and the predicted classes on the horizontal axes.

Eating/Drinking

This section focuses on the results obtained for the Eating/Drinking activity detection. The overall accuracy using the aforementioned model was ~74%. As Table 2 shows, by using a combination of HOG features collected on an input image scaled to 3:2, 2:3, and 1:3, a class accuracy of ~90% was obtained.

Qualitative examples of the true positives, false positives, and false negatives are shown in Figure 14. The false positives occurred when the hand was close to the mouth, which tricked the classifier into “thinking” that the driver was eating/drinking. Using a cellphone of a long length in a texting position also caused misclassification. False negatives were those that were missed by the detector; as explained earlier, choosing the right slack parameter can help minimize this type of error.

Table 2. Confusion matrix for the Eating/Drinking action classification.

	Eating/ Drinking	Talking	Visor	Center Stack	Texting	One Hand	Both Hands	Misc. 1	Misc. 2
Eating/ Drinking	90.5%	8.5%	0.0%	0.5%	0.2%	0.0%	0.3%	0.0%	0.0%
Talking	16.6%	82.9%	0.1%	0.0%	0.4%	0.0%	0.0%	0.0%	0.0%
Visor	30.3%	16.0%	50.9%	0.0%	1.5%	0.5%	0.7%	0.0%	0.0%
Center Stack	4.8%	2.3%	0.0%	92.1%	1.5%	0.5%	0.7%	0.0%	0.0%
Texting	20.0%	0.0%	0.0%	0.2%	43.7%	25.6%	9.8%	0.6%	0.0%
One Hand	0.0%	0.0%	0.0%	14.0%	0.5%	85.1%	0.3%	0.0%	0.0%
Both Hands	0.2%	0.0%	0.0%	0.0%	15.4%	10.3%	74.1%	0.0%	0.0%
Misc. 1	0.9%	0.0%	0.0%	0.9%	16.9%	6.3%	68.8%	6.3%	0.0%
Misc. 2	0.0%	0.0%	0.0%	11.6%	0.0%	42.3%	46.2%	0.0%	0.0%

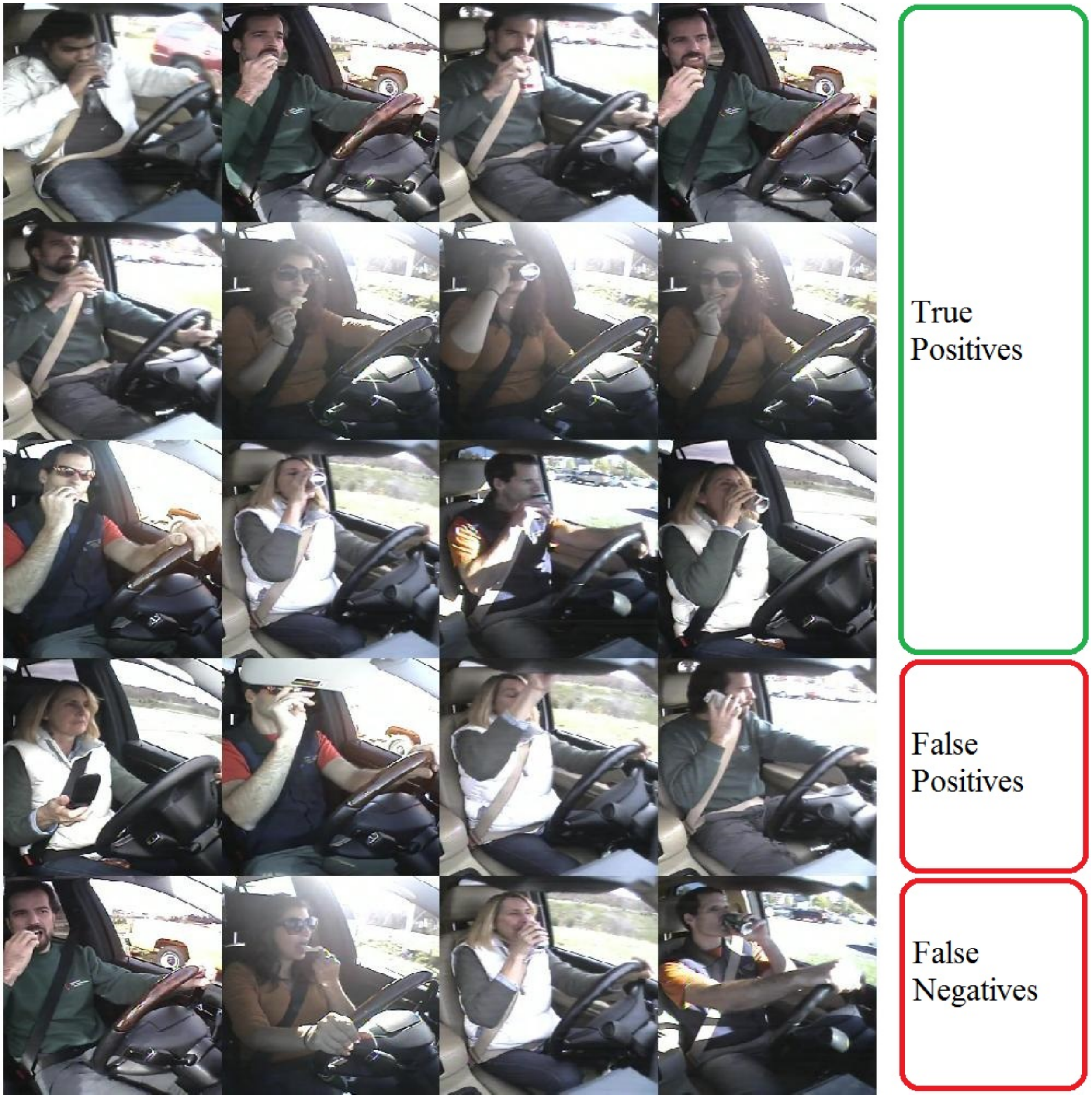

Figure 14. Images. Qualitative examples of a model fine-tuned for Eating/Drinking action classification.

Talking

This section will discuss the results obtained for the Talking activity detection. The overall accuracy for detecting this activity was ~73%. A class accuracy of ~85% was obtained on the test data set. Drivers in the false positive cases had their hands very close to their ear or did not have their hand in the crop generated for the data set. Qualitative examples of true positives, false positives, and false negatives are shown in Figure 15, and quantitative results are shown in Table 3.

Table 3. Confusion matrix for the Talking action classification.

	Eating/ Drinking	Talking	Visor	Center Stack	Texting	One Hand	Both Hands	Misc. 1	Misc. 2
Eating/ Drinking	85.9%	7.4%	3.1%	2.1%	0.9%	0.0%	0.3%	0.1%	0.0%
Talking	15.1%	84.9%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Visor	32.0%	18.91%	44.2%	0.0%	1.2%	1.2%	0.0%	2.5%	0.0%
Center Stack	1.0%	5.2%	0.0%	89.2%	4.2%	0.35%	0.0%	0.0%	0.0%
Texting	15.5%	0.1%	0.0%	9.1%	43.7%	25.0%	5.3%	0.0%	1.3%
One Hand	0.0%	0.0%	0.0%	11.9%	0.2%	87.4%	0.2%	0.1%	0.1%
Both Hands	0.2%	0.0%	0.0%	0.0%	11.4%	5.5%	72.9%	9.8%	0.3%
Misc. 1	0.9%	0.0%	0.9%	0.9%	13.4%	3.6%	41.1%	32.1%	7.1%
Misc. 2	0.0%	0.0%	0.0%	11.6%	0.0%	23.0%	19.2%	11.5%	34.6%

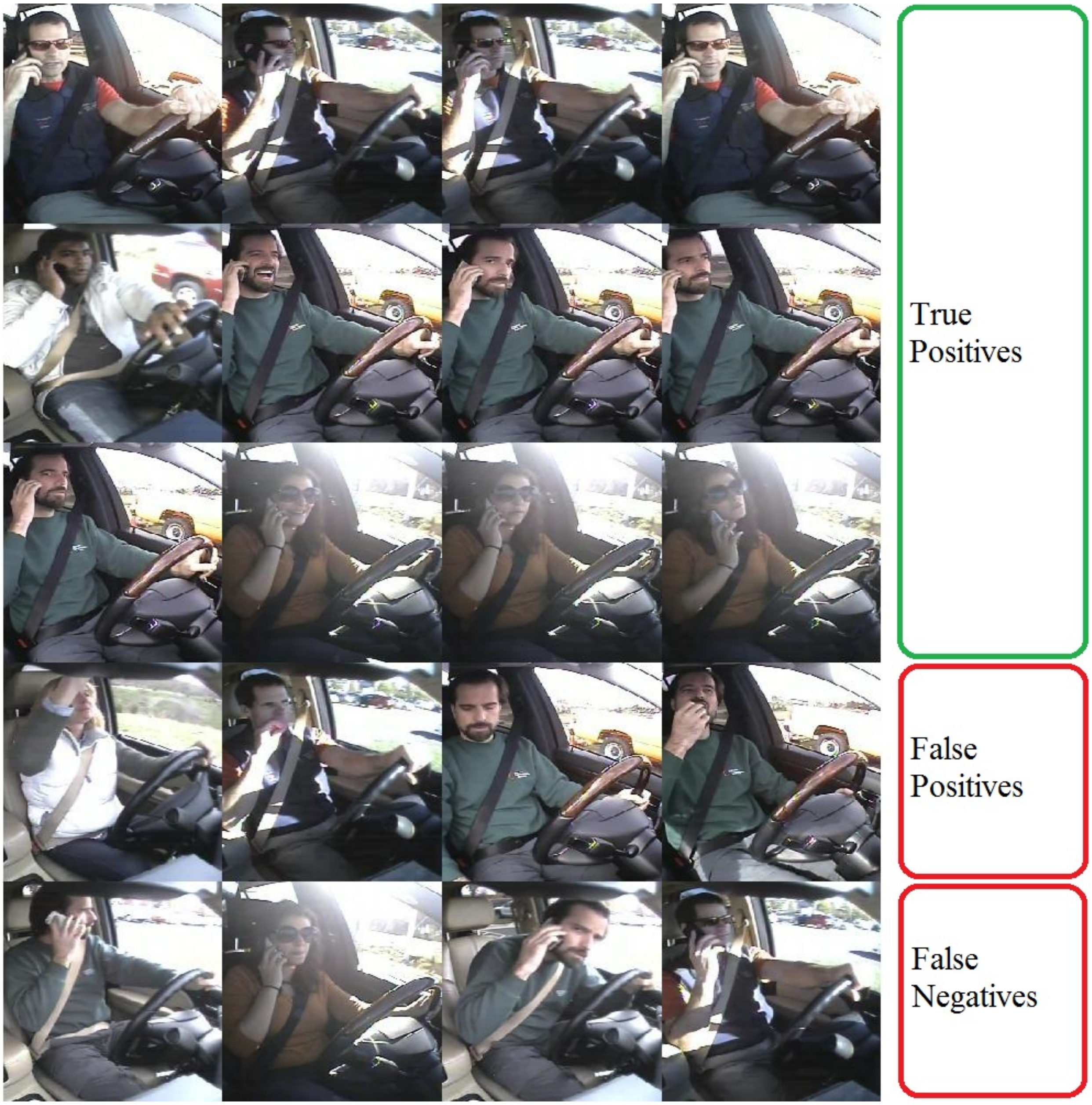

Figure 15. Images. Qualitative examples of a model fine-tuned for Talking activity classification.

Visor

This section focuses on the results obtained for the Visor activity detection. An accuracy of 83% was obtained for this classifier; overall accuracy was ~72%. For this class, only 19 participants were used for training, as adding additional data decreased the accuracy by 4%. The test set had six participants instead of five. The drop in accuracy is a common problem in training machine-learning algorithms, referred to as "bias-variance tradeoff." Adding additional data from the 20th participant increased the variance of the training data and resulted in a loss of generalization and accuracy. Qualitative examples of true positives, false positives, and false negatives are shown in Figure 16, and quantitative results are shown in Table 4.

Table 4. Confusion matrix for the Visor classification.

	Eating/ Drinking	Talking	Visor	Center Stack	Texting	One Hand	Both Hands	Misc. 1	Misc. 2
Eating/ Drinking	67.5%	21.4%	9.3%	0.0%	0.6%	0.0%	1.0%	0.0%	0.0%
Talking	35.7%	58.5%	5.7%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Visor	9.5%	6.7%	83.0%	0.0%	0.4%	0.0%	0.1%	0.0%	0.0%
Center Stack	10.2%	1.7%	0.0%	84.3%	0.7%	2.6%	0.2%	0.0%	0.0%
Texting	20.9%	0.2%	0.4%	3.1%	47.9%	19.5%	7.7%	0.0%	0.0%
One Hand	4.6%	2.0%	2.9%	4.5%	3.4%	81.2%	1.0%	0.0%	0.0%
Both Hands	6.0%	0.24%	3.0%	0.0%	8.3%	2.0%	80.3%	0.0%	0.0%
Misc. 1	1.7%	0.0%	3.5%	0.9%	17.6%	7.0%	64.6%	5.3%	0.0%
Misc. 2	0.0%	0.0%	0.0%	23.0%	0.0%	30.7%	46.2%	0.0%	0.0%

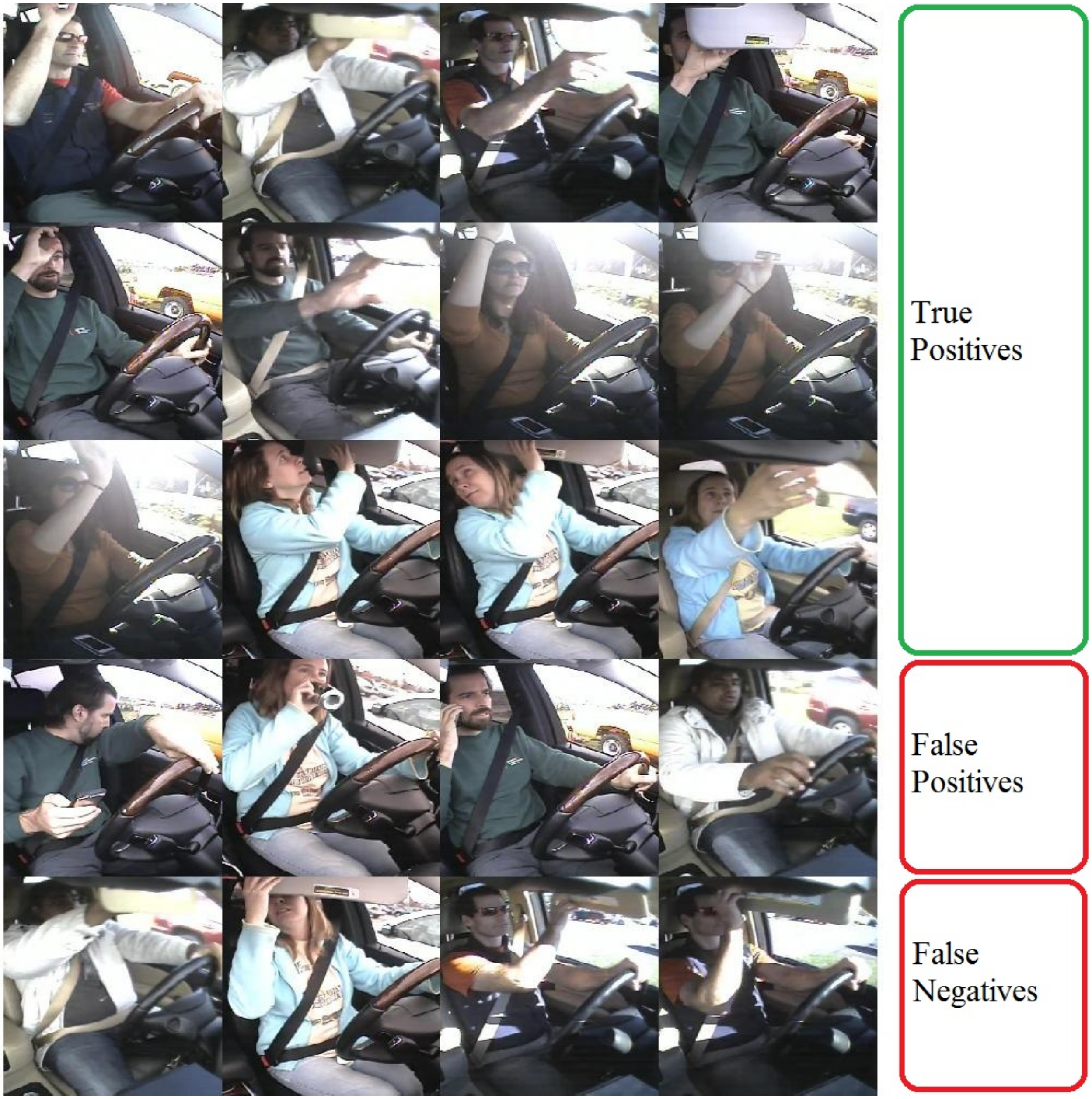

Figure 16. Images. Qualitative examples of a model fine-tuned for Visor activity classification.

Center Stack

In this section, we present the results obtained for the Center-Stack activity detection. Accuracy in detecting this activity was ~92%. Overall accuracy was ~71%. Again, the limitation of the method is clear—if the limbs are obscured or not completely present in the image crop, accuracy will suffer. Qualitative examples of true positives, false positives, and false negatives are shown in Figure 17, and quantitative results are shown in Table 5.

Table 5. Confusion matrix for the Center-Stack action classification.

	Eating/ Drinking	Talking	Visor	Center Stack	Texting	One Hand	Both Hands	Misc. 1	Misc. 2
Eating/ Drinking	82.7%	10.2%	3.2%	2.3%	1.1%	0.0%	0.3%	0.0%	0.0%
Talking	21.9%	77.9%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Visor	33.6%	17.7%	46.7%	0.0%	1.0%	0.1%	0.7%	0.0%	0.0%
Center Stack	2.0%	5.1%	0.0%	92.4%	0.0%	0.3%	0.0%	0.0%	0.0%
Texting	22.0%	0.0%	0.1%	6.6%	43.0%	25.1%	3.1%	0.0%	0.0%
One Hand	0.1%	0.0%	0.0%	11.9%	3.7%	83.8%	0.1%	0.0%	0.0%
Both Hands	0.3%	0.0%	0.0%	0.1%	11.7%	13.1%	74.5%	0.0%	0.0%
Misc. 1	0.9%	0.0%	0.0%	0.9%	15.1%	6.3%	58.9%	17.8%	0.0%
Misc. 2	0.0%	0.0%	0.0%	30.7%	0.0%	23.0%	46.2%	0.0%	0.0%

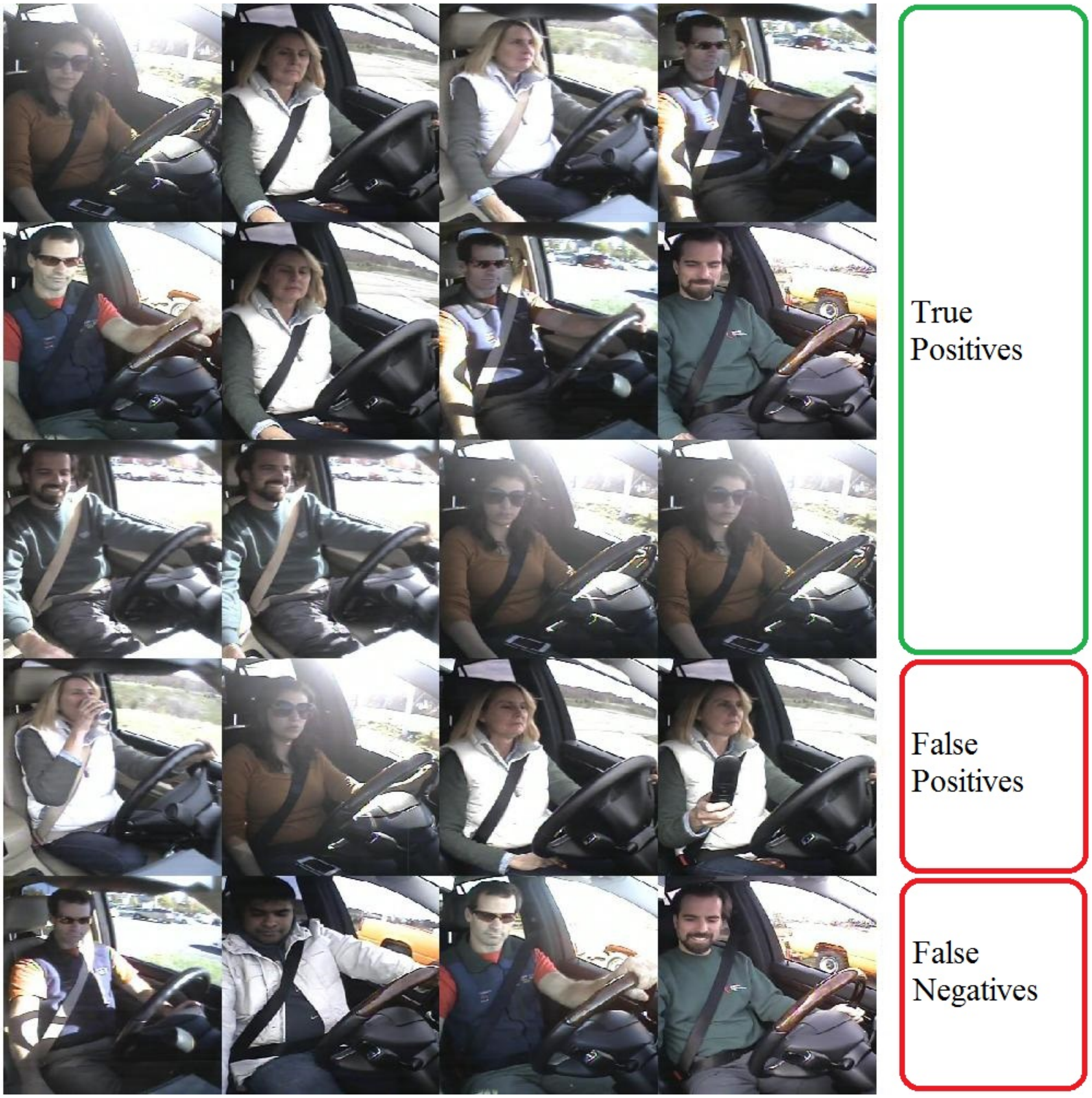

Figure 17. Images. Qualitative examples of a model fine-tuned for Center-Stack activity classification.

Texting

This section discusses the results obtained for the Texting activity detection. Of the seven classifiers, this one was the least accurate. Low accuracy can be attributed to the extreme variation in pose across people and the shape of the device being used. An accuracy of ~56% was achieved with this classifier. An overall accuracy of ~71% was obtained. In the current setup, there is a very small difference in the position of the hand being used for texting due to foreshortening effects. An overhead camera, placed directly above the driver, could magnify the hand position by considerably reducing the amount of foreshortening and therefore improve accuracy. The other significant reason for the confusion between the Texting class and the One-Hand class is the location of the participant's right hand. The mean and standard deviation of the *x* and *y* coordinates of the limbs were 293 pixels, 66 pixels, 149 pixels, and 52 pixels, respectively, for the Texting class. For the One-Hand class, the mean and standard deviation of the *x* and *y* coordinates of the limbs were 302 pixels, 55 pixels, 68 pixels, and 55 pixels, respectively. This overlap added to the confusion between the classes. A deep-learning-based method would be able to reason various nuances of this class better. Qualitative examples of the true positives, false positives, and false negatives are shown in Figure 18, and quantitative results are shown in Table 6.

Table 6. Confusion matrix for the Texting action classification.

	Eating/ Drinking	**Talking**	**Visor**	**Center Stack**	**Texting**	**One Hand**	**Both Hands**	**Misc. 1**	**Misc. 2**
Eating/ Drinking	73.7%	14.2%	7.1%	0.0%	2.4%	0.0%	2.3%	0.0%	0.0%
Talking	34.0%	65.1%	0.7%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%
Visor	18.6%	12.3%	67.2%	0.0%	0.1%	0.0%	0.1%	1.4%	0.0%
Center Stack	4.0%	10.0%	0.0%	84.1%	0.7%	0.7%	0.0%	0.0%	0.0%
Texting	3.6%	3.7%	0.8%	0.0%	55.9%	22.3%	0.4%	12.2%	0.9%
One Hand	2.9%	0.0%	0.0%	1.4%	0.0%	93.4%	0.3%	1.6%	0.0%
Both Hands	0.0%	0.0%	0.1%	0.0%	13.0%	0.7%	58.3%	26.8%	0.8%
Misc. 1	0.9%	0.0%	0.0%	0.0%	5.3%	0.0%	48.2%	42.8%	2.6%
Misc. 2	0.0%	0.0%	0.0%	0.0%	0.0%	38.4%	34.6%	3.8%	23.0%

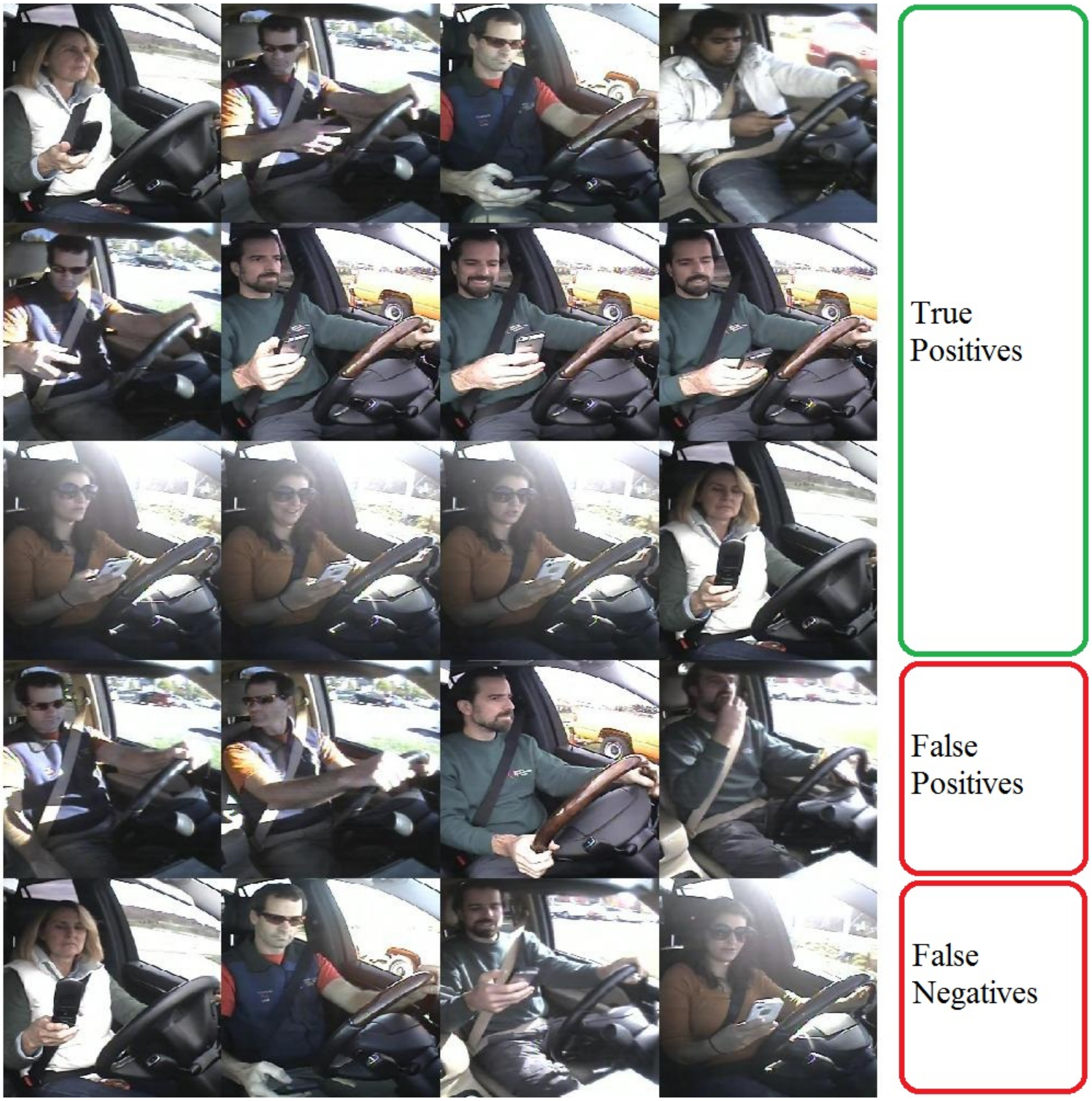

Figure 18. Images. Qualitative examples of a model fine-tuned for Texting activity classification.

One Hand on the Wheel

Results obtained for the One Hand on the Wheel activity detection are presented in this section. An accuracy of ~96% was achieved for this class. The overall accuracy was ~70%. The high accuracy can be attributed to the fact that there was a very small amount of variation (participants' hands were on their laps) compared to the other classes. Qualitative examples of true positives, false positives, and false negatives are shown in Figure 19, and quantitative results are shown in Table 7.

Table 7. Confusion matrix for the One Hand on the Wheel action classification.

	Eating/ Drinking	Talking	Visor	Center Stack	Texting	One Hand	Both Hands	Misc. 1	Misc. 2
Eating/ Drinking	80.2%	14.7%	1.5%	0.6%	0.7%	0.0%	1.8%	0.3%	0.0%
Talking	44.5%	55.0%	0.1%	0.0%	0.3%	0.0%	0.0%	0.0%	0.0%
Visor	38.3%	15.9%	41.3%	0.3%	1.6%	0.0%	1.3%	0.9%	0.0%
Center Stack	5.84%	1.0%	0.0%	90.2%	2.4%	0.2%	0.0%	0.0%	0.0%
Texting	13.2%	0.0%	0.1%	0.8%	48.8%	20.9%	12.3%	3.6%	0.1%
One Hand	0.1%	0.0%	0.0%	2.9%	0.0%	96.2%	0.1%	0.2%	0.0%
Both Hands	0.0%	0.0%	0.0%	0.0%	6.4%	2.7%	75.1%	15.2%	0.3%
Misc. 1	0.9%	0.0%	0.0%	0.0%	6.2%	2.6%	43.7%	39.2%	7.1%
Misc. 2	0.0%	0.0%	0.0%	0.0%	0.0%	50.0%	46.2%	3.8%	0.0%

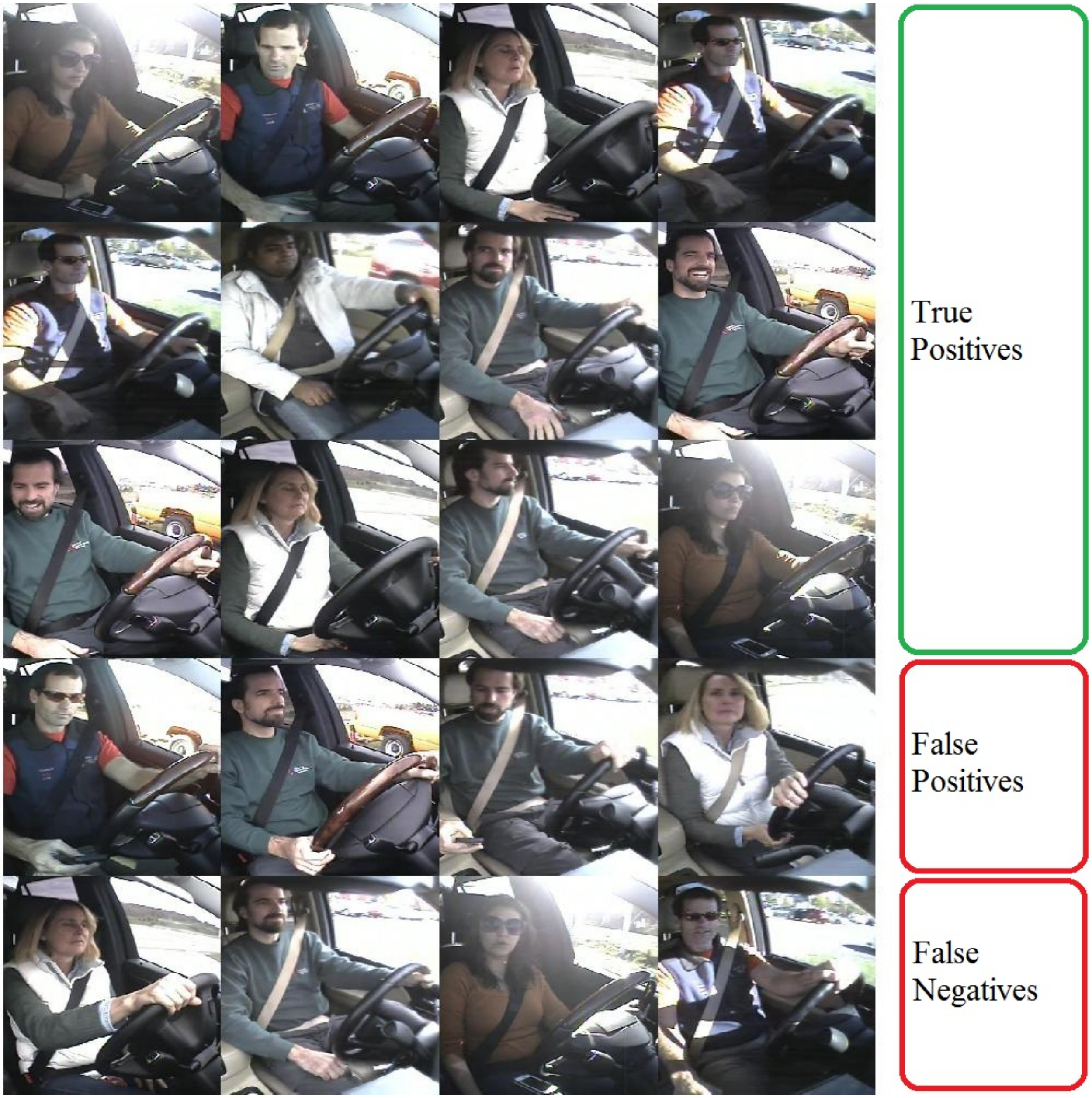

Figure 19. Images. Qualitative examples of a model fine-tuned for One Hand on the Wheel activity classification.

Both Hands on the Wheel

This section discusses the results obtained for the Both Hands on the Wheel activity detection. An accuracy of ~88% was obtained for this class. An overall accuracy of ~71% was obtained. The variation in the placement of hands on the steering wheel can be quite large, and this caused the model to fail occasionally. This can be addressed by increasing the diversity of the training data to capture extreme variations. Qualitative examples of true positives, false positives, and false negatives are shown in Figure 20, and quantitative results are shown in Table 8.

Table 8. Confusion matrix for the Both Hands on the Wheel action classification.

	Eating/ Drinking	Talking	Visor	Center Stack	Texting	One Hand	Both Hands	Misc. 1	Misc. 2
Eating/ Drinking	76.14%	14.2%	7.9%	0.1%	0.1%	0.0%	1.4%	0.0%	0.0%
Talking	48.3%	39.5%	12.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Visor	28.4%	7.9%	61.0%	0.0%	0.1%	0.0%	2.3%	0.0%	0.0%
Center Stack	5.4%	0.8%	0.3%	92.2%	0.0%	0.9%	0.0%	0.0%	0.0%
Texting	19.7%	7.8%	4.2%	0.5%	41.0%	25.7%	0.9%	0.6%	0.0%
One Hand	0.0%	4.1%	3.1%	0.3%	0.1%	91.7%	0.3%	0.0%	0.0%
Both Hands	0.2%	0.0%	0.0%	0.0%	7.3%	2.0%	88.3%	1.9%	0.0%
Misc. 1	0.9%	0.0%	0.8%	0.0%	10.7%	6.3%	72.3%	8.9%	0.0%
Misc. 2	0.0%	7.6%	0.0%	15.3%	0.0%	19.2%	57.6%	0.0%	0.0%

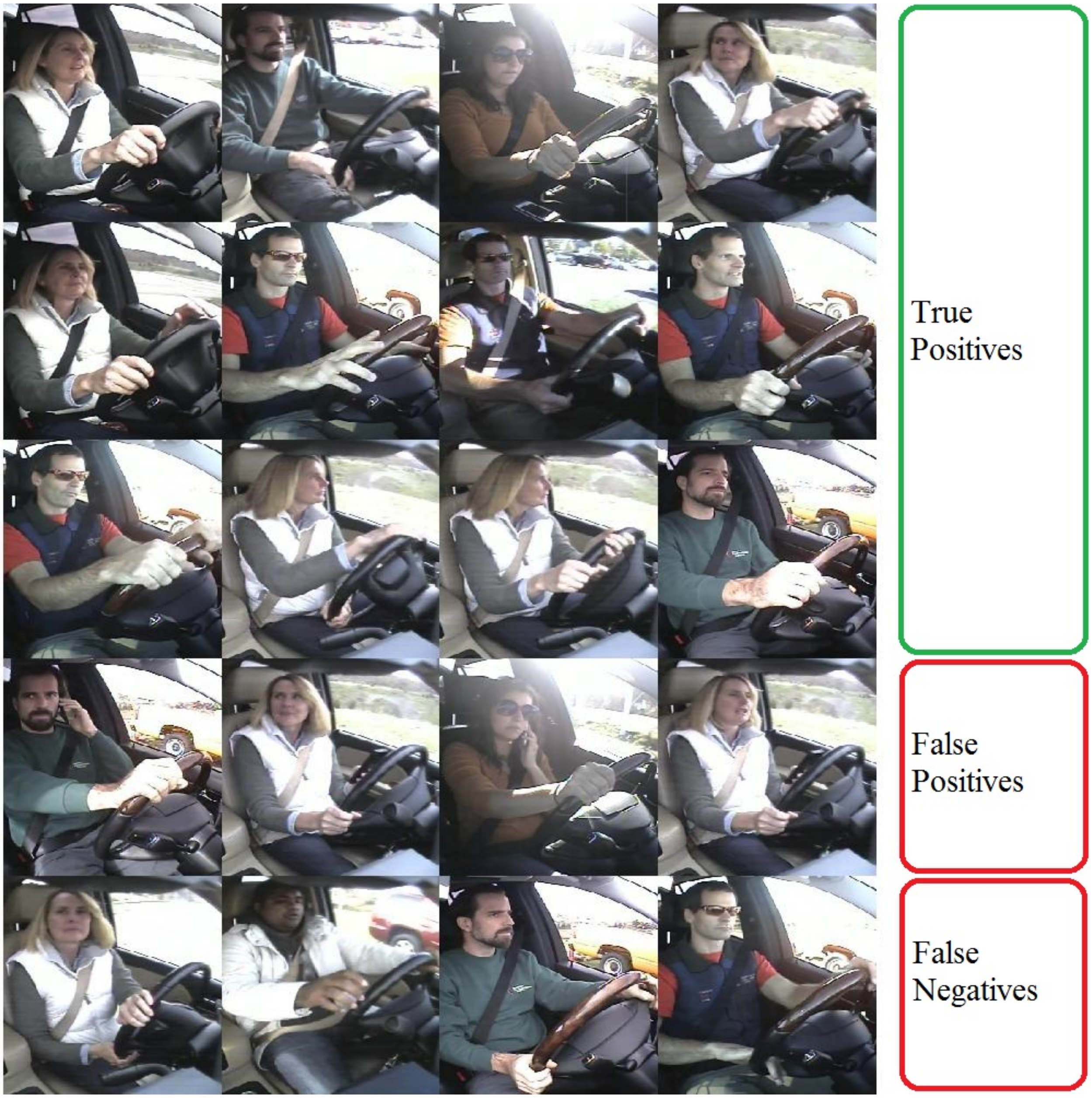

Figure 20. Images. Qualitative examples of a model fine-tuned for Both Hands on the Wheel activity classification.

CHAPTER 4. CONCLUSIONS AND FUTURE WORK

This report demonstrated the efficacy of computer-vision and machine-learning-based approaches in a real-world environment applied for activity recognition using three different methods. The goal of automating the reduction of the driver's pose was accomplished and much experience gained during the process. The researchers performed extensive testing of the methods, improved them, and contributed to original research in the area of computer vision with the asymmetric downsampling method.

This report also describes various methods that did not work and also proposes methods that will work better than the methods implemented in the given time frame.

The accuracy of the Texting activity class recognition is still low compared to the other classes due to the large variation in the hand positions of people holding their cellphones while performing texting activities. This is unfortunate, as this activity is of particular interest because of its association with crash risk (Fitch et al., 2013). However, there are a variety of research questions wherein the class predictions at the accuracy levels achieved would be beneficial, including research questions based on probability and risk measurements. Furthermore, the class-specific models can be used together in a decision tree framework to obtain a higher overall accuracy for any specific use case. By classifying the images using the most accurate model first and cascading through the rest of the models, accuracy improvements with the overall classifier can be realized.

To capture all available information, a wide-angle lens placed near the visor could provide a better camera angle with a better view of the driver's upper body, and thus improve the accuracy of machine-learning models. Putting a scalable system in place with large amounts of labeled data available to work with would also improve accuracy. The researchers predict that with twice as much data, an additional 10% increase in accuracy can be obtained. After the initial reduction for obtaining training data, these methods may completely eliminate the necessity for human effort in identifying driver activities.

There are a variety of implementations of deep learning methods that can be used for activity detection. The researchers used the implementation of Krizhevsky, Sutskever, and Hinton (2012), which is a deep convolutional neural network designed to classify images. The researchers used data augmentation to solve the driver activity classification with an 83% accuracy within a short 10-week phase. Data augmentation was used to increase the amount of training data in a short time without manual reduction. The images were subjected to minor perturbations, which produces multiple copies of the images with minor differences, such as translation, rotation, and color channel perturbation. A variety of deep neural networks were evaluated during this phase, which included tuning various parameters of the networks and the network itself. Even after augmenting the data by 45 times (50,000 images were used to generate 2.3 million images), the models were still overfitting. To address this shortcoming, the authors suggest the construction of a deep neural network along the lines of Krizhevsky et al. (2012) with custom data augmentation to build driver pose estimation and activity recognition models that reason about various deeper nuances in a fashion similar to the human visual cortex. Human engineered features are no longer needed as these models “learn” the necessary features themselves. Human engineered features will always have a place among machine-learning methods, but to capture the variations (lighting, color, shape, etc.) involved in the real world, these features become problematically complex and problem specific. Deep learning offers a more holistic approach to classic computer vision problems involving exponentially large variations.

Throughout the course of this study, researchers gained a great deal of experience in applying various machine-learning techniques and examining their failure and success modes. As data scientists can work with any problem type and can view data objectively, the same methods can be applied to a wide variety of problems, such as pedestrian detection, traffic sign identification, etc., providing an even greater benefit in the field of traffic safety research.

APPENDIX A. INFORMED CONSENT FORM

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Informed Consent for Participants of Investigative Projects

Pose Estimation and Activity Recognition of Drivers

Investigators: Zac Doerzaph, Loren Stowe, Ujwal Krothapalli and Abhijit Sarkar: Virginia Tech Transportation Institute (VTTI)

I. THE PURPOSE OF THIS RESEARCH PROJECT

The purpose is to develop automated means of analyzing video data to classify the actions of drivers.

We will be collecting video data from 30 drivers. Pose information of the driver could provide measures of the hand position (e.g. steering wheel, center stack, etc.) thus, augmenting and possibly replacing costly manual data reduction processes. Machine learning is a tool that can be used to mimic the human decision making process in the form of algorithms. Combined with image feature extraction methods these algorithms can 'learn' the patterns (poses) present in the data. The lead researchers believe these technologies can be applied to provide the position of the driver's hands or other body parts. The efficacy of such methods has been proven on a sample dataset. To be able to train these algorithms, datasets have to be created manually labeling various parts of the driver such as, head and the arms. About 180 frames per driver (in various poses) need to be reduced manually to help train and validate the pose estimation algorithms. The research will generate a new dataset consisting of 30 drivers through manual reduction and further improve the accuracy of the pose estimating algorithms.

II. PROCEDURES

During the course of this experiment, you will be asked to perform the following tasks:

1. Read this Informed Consent Form and sign it if you agree to participate.
2. Show the experimenter your valid driver's license.
3. Inform the experimenter if you have any food or beverage allergies.

All participants will be asked to complete Session 1. If the Session 2 box is checked below, you will also be asked to complete Session 2

Session 1 (30 minutes)

1. Sit in the vehicle and perform the following actions as per the experimenter's instructions.
2. These actions are to be performed for about 2 minutes each, when the car is parked. Video data will be collected of you while you are in the vehicle. The experimenter will provide you with a selection of snacks, beverages and other instruments such as a cellphone and let you know when to switch tasks. The experimenter will then ask you to perform the same tasks in a second car with a different interior.

- Eating
- Talking on the phone (phone against the ear)
- Drinking a beverage
- Hands on the center stack
- Texting
- Hand on the visor
- Both hands on the wheel
- One hand on wheel

Session 2 (30 Minutes)

You will be asked to drive the vehicle on the VTTI Smart Road, adhering to the 35 mph speed limit. The experimenter will ask you to repeat the tasks above while driving. You may opt out of a task at any time.

It is important for you to understand that we are not evaluating you or your performance in any way. The data you provide will help us refine the VTTI software, and this is very important to our future project. Approximately 30 participants will participate in this study.

III. RISKS

Parking Lot Session

The risk to you during the static testing session in the parking lot is that of sitting in a parked vehicle and moving your limbs. This study involves snacks and a drink, there is a risk that you could be allergic to one of the ingredients if you have any food allergies.

Smart Road Session

The risk to those that participate in the Smart Road session is that of an accident normally present when driving on a closed road (VTTI smart road), and the risk involved when driving an unfamiliar vehicle performing tasks that are a part of your normal driving behavior.

While the risk of participation in this study is considered to be no more than that encountered in everyday driving, if you are pregnant you should talk to your physician and discuss this consent form with them before making a decision about participation.

In the event of an accident or injury in an automobile owned or leased by Virginia Tech, the automobile liability coverage for property damage and personal injury is provided. The total policy amount per occurrence is $2,000,000. This coverage (unless the other party was at fault, which would mean all expense would go to the insurer of the other party's vehicle) would apply in case of an accident for all volunteers and would cover medical expenses up to the policy limit. For example, if you were injured in an automobile owned or leased by Virginia Tech, the cost of transportation to the hospital emergency room would be covered by this policy.

Participants in a study are considered volunteers, regardless of whether they receive payment for their participation; under Commonwealth of Virginia law, worker's compensation does not apply to volunteers; therefore, if not in the automobile, the participants are responsible for their own medical insurance for bodily injury. Appropriate health insurance is strongly recommended to cover these types of expenses. For example, if you were injured outside of the automobile owned or leased by Virginia Tech, the cost of transportation to the hospital emergency room would be covered by your insurance.

The following precautions will be taken to ensure minimal risk to you:

1. You may take breaks or decide to cease participation at any time.
2. The vehicle is equipped with a driver’s side and passenger’s side airbag supplemental restraint system.
3. You are required to wear your lap and shoulder belt restraint system while in the vehicle.
4. In the event of a medical emergency, or at your request, VTTI staff will arrange medical transportation to a nearby hospital emergency room. You may elect to undergo examination by medical personnel in the emergency room.
5. All data collection equipment is mounted such that, to the greatest extent possible, it does not pose a hazard to you in any foreseeable case. It does not interfere with any part of your normal field of view. The addition of the system to the vehicle will in no way affect the operating or handling characteristics of the vehicle.
6. Participation or nonparticipation in this study will not impact your employment at VTTI.

IV. BENEFITS

While there are no direct benefits to you from this research, you may find the experiment interesting. No promise or guarantee of benefits is made to encourage you to participate. Participation in this study will be used to build a driver pose estimation model.

V. EXTENT OF ANONYMITY AND CONFIDENTIALITY

The data gathered in this experiment will be treated with confidentiality. Shortly after participation, your name will be separated from your data. A coding scheme will be employed to identify the data by participant number only (e.g., Participant No. 1). You may elect to have your data withdrawn from the study if you so desire, but you must inform the experimenters immediately of this decision so that the data may be promptly removed.

VTTI researchers will not release data identifiable to an individual to anyone other than VTTI staff without your written consent. With your permission, VTTI researchers may show specific clips of video at research conferences and for research demonstration purposes. The data collected in this study may be used in future VTTI transportation research projects. IRB approval will be obtained prior to accessing the data for other projects.

It is possible that the Institutional Review Board (IRB) may view this study's collected data for auditing purposes. The IRB is responsible for the oversight of the protection of human subjects involved in research.

COMPENSATION

No compensation will be provided. This study must take place on your own time.

FREEDOM TO WITHDRAW

As a participant in this research, you are free to withdraw at any time without penalty.

VIII. APPROVAL OF RESEARCH

This research project has been approved, as required, by the Institutional Review Board for Research Involving Human Subjects at Virginia Polytechnic Institute and State University. This approval is valid through the date listed at the bottom of this form.

IX. PARTICIPANT'S RESPONSIBILITIES

If you voluntarily agree to participate in this study, you will have the following responsibilities:

1. To inform the experimenter if you have difficulties of any type.
2. To wear your seat and lap belt at all times while operating the vehicle.
3. To operate the vehicle in a safe manner
4. To observe all traffic laws applying to the area in which the vehicle is being operated.
5. To inform the experimenter if you have any food allergies.

X. PARTICIPANT'S ACKNOWLEDGMENTS

Please check one of the following:

☐ VTTI **has my permission** to use the digital video including my image for research demonstration and research presentation purposes.

☐ VTTI **does not have my permission** to use the digital video including my image for research demonstration and research presentation purposes. I understand that VTTI will maintain possession of the video, and only use it for research purposes.

Check all that apply:

☐ I am not under the influence of any substances or taking any medications that may impair my ability to participate safely in this experiment.

☐ I have informed the experimenter of any concerns/questions I have about this study.

☐ I understand that digital video including my image and audio will be collected as part of this experiment.

□ If I am pregnant, I acknowledge that I have either discussed my participation with my physician, or that I accept any additional risks due to pregnancy.

.

XI. PARTICIPANT'S PERMISSION

I have read and understand the Informed Consent and conditions of this project. I have had all my questions answered. I hereby acknowledge the above and give my voluntary consent for participation in this project. **If I participate, I may withdraw at any time without penalty. I agree to abide by the rules of this project, as well as the policies for vehicle use.**

____

Participant's name (Print) Signature Date

____

Researcher's name (Print) Signature Date

===

Should I have any questions about this research or its conduct, I may contact:

Zac Doerzaph	Principal Investigator
Ujwal Krothapalli	Investigator

If I should have any questions about the protection of human research participants regarding this study, I may contact:

David Moore Chair, Virginia Tech Institutional Review (540) 231-4991
moored@vt.edu Board for the Protection of Human Subjects
Office of Research Compliance

Blacksburg, VA 24060

The Participant Must Be Provided With A Copy Of This Consent Form.

APPENDIX B. PARTICIPANT RECRUITMENT ADVERTISEMENT

The Connected and Advanced Vehicle Systems (CAVS) group at VTTI is currently recruiting participants for a study to validate algorithms that estimate driver body pose and classify the different actions using Machine-learning based methods. This is the first study of its kind at VTTI and we would be glad to recruit volunteers. We cannot provide any compensation and the study would take about 30 to 60 minutes depending on your level of participation. The timing is flexible and we can accommodate you anytime between 7AM and 6PM on weekdays. A video of your face and arms will be recorded, you have the option of preventing these videos from being used in presentations and only for research.

We are looking for about 30 participants and 10 out of the 30 will be asked to drive on the smart road. The participants will be asked to perform a fixed set of actions (eating, drinking, hand on the center stack etc.) with their arms in a parked car and on the smart road (optional). The participation would have to be on your own time (it is not part of your employment at VTTI), but we can provide a variety of snacks and beverages. We are looking a diverse range of people to include in this study, would you be interested in being a participant?

Please contact Ujwal K at @vtti.vt.edu if you are interested or have any questions.

Note: Participation, or lack of participation will have no influence on your employment at VTTI.

APPENDIX C. PARTICIPANT SCREENING FORM

Screening Form

1. Are you currently employed at VTTI?
    a. Yes
    b. No (Not eligible)
2. Are you at least 18 years of age or older?
    a. Yes
    b. No (not eligible)
3. Do you have a valid driver’s license? (*Mention that we will need to present proof of a valid license when they sign the consent form*)
    a. Yes
    b. No (not eligible)
4. Are you able to operate an automatic transmission without assistive devices?
    a. Yes
    b. No (not eligible)
5. Do you suffer from any food or drink allergies?
    a. Yes
    b. No

REFERENCES

Andriluka, M., Roth, S., & Schiele, B. (2009). *Pictorial structures revisited: People detection and articulated pose estimation.* Paper presented at the IEEE Conference on Computer Vision and Pattern Recognition, 2009.

Bourdev, L., & Malik, J. (2009). *Poselets: Body part detectors trained using 3d human pose annotations.* Paper presented at the 2009 IEEE 12th International Conference on Computer Vision.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST), 2*(3), 27.

Dalal, N., & Triggs, B. (2005). *Histograms of oriented gradients for human detection.* Paper presented at the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.

Dingus, T. A., & Klauer, S. G. (2008). *The relative risks of secondary task induced driver distraction* (SAE Technical Paper 21-0001). Warrendale, PA: Society of Automotive Engineers.

Eichner, M., Ferrari, V., & Zurich, S. (2009). Better appearance models for pictorial structures. In *British Machine Vision Conference (BMVC)*. Retrieved from https://www.research.ed.ac.uk/portal/files/17738266/Eichner_Ferrari_2009_Better_appearance_models.pdf

Everts, I., van Gemert, J. C., & Gevers, T. (2013). *Evaluation of color stips for human action recognition.* Paper presented at the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(9), 1627-1645.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision, 61*(1), 55-79.

Ferrari, V., Marin-Jimenez, M., & Zisserman, A. (2008). *Progressive search space reduction for human pose estimation.* Paper presented at the IEEE Conference on Computer Vision and Pattern Recognition, 2008.

Fitch, G. M., Soccolich, S. A., Guo, F., McClafferty, J., Fang, Y., Olson, R. L., ... & Dingus, T. A. (2013). *The impact of hand-held and hands-free cell phone use on driving performance and safety-critical event risk* (No. DOT HS 811 757). Washington, DC: National Highway Traffic Safety Administration.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *Imagenet classification with deep convolutional neural networks.* In *Advances in neural information processing systems* (pp. 1097-1105).

Krothapalli, U., & Christie, G. (Producer). (2013). *Gesture activated interactive assistant.* Retrieved from https://www.youtube.com/watch?v=VFPAHY7th9A

Mori, G., & Malik, J. (2002). Estimating human body configurations using shape context matching. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen (Eds.), *Computer vision: Proc. 7th European Conf. on Computer Vision (ECCV 2002), Part III*. Berlin, Germany: Springer-Verlag.

O'Rourke, J., & Badler, N. (1980). Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 6*, 522-536.

Raptis, M., & Sigal, L. (2013). *Poselet key-framing: A model for human activity recognition.* Paper presented at the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Rohr, K. (1994). Towards model-based recognition of human movements in image sequences. *CVGIP: Image understanding, 59*(1), 94-115.

Sapp, B., Toshev, A., & Taskar, B. (2010). Cascaded models for articulated pose estimation. In Daniilidis K., Maragos P., Paragios N. (eds.), *Computer Vision–ECCV 2010* (pp. 406-420). Springer-Verlag Berlin Heidelberg.

Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., . . . Moore, R. (2013). Real-time human pose recognition in parts from single depth images. *Communications of the ACM, 56*(1), 116-124.

Sigal, L., & Black, M. J. (2006). *Measure locally, reason globally: Occlusion-sensitive articulated pose estimation.* Paper presented at the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

Singh, V. K., Nevatia, R., & Huang, C. (2010). Efficient inference with multiple heterogeneous part detectors for human pose estimation. In Daniilidis K., Maragos P., Paragios N. (eds.), *Computer Vision–ECCV 2010* (pp. 314-327). Springer-Verlag Berlin Heidelberg.

Sullivan, J., & Carlsson, S. (2002). Recognizing and tracking human action. In Heyden A., Sparr G., Nielsen M., Johansen P. (eds.), *Computer Vision—ECCV 2002* (pp. 629-644). Springer-Verlag Berlin Heidelberg.

Sun, M., & Savarese, S. (2011). *Articulated part-based model for joint object detection and pose estimation.* Paper presented at the 2011 IEEE International Conference on Computer Vision (ICCV).

Tian, Y., Sukthankar, R., & Shah, M. (2013). *Spatiotemporal deformable part models for action detection.* Paper presented at the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Tsochantaridis, I., Hofmann, T., Joachims, T., & Altun, Y. (2004). *Support vector machine learning for interdependent and structured output spaces.* Paper presented at the Proceedings of the Twenty-First International Conference on Machine Learning.

Ullman, S. (2009). From classification to full object interpretation. In Dickinson, S., Leonardis, A., Schiele, B., & Tarr, M. (eds.), *Object Categorization: Computer and Human Vision Perspectives*, (pp. 288-300). New York: Cambridge University Press.

Wang, Y., Tran, D., Liao, Z., & Forsyth, D. (2012). Discriminative hierarchical part-based models for human parsing and action recognition. *The Journal of Machine Learning Research, 13*(1), 3075-3102.

Yang, Y., & Ramanan, D. (2013). Articulated human detection with flexible mixtures of parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(12), 2878-2890.