Optical Flow for De-Identification and Driver Behavior Classification

Arjun Sarup

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2021-188
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-188.html

August 13, 2021

Acknowledgement

A special shoutout to Professor Friedland, who made it his personal mission to shine away any difficulty that came my way. Thank you to Dr. Meredith M. Lee for teaching me what leadership, tenacity and collaboration actually means. To the best parents in the world — my Mom and Dad. Thank you for being the Genies who made all of my wishes come true, for conjuring up the tallest trees of love and hope with your magic beans. All the hat-tips in the world to my Nanu, who never failed to pick me up. My Nani — for teaching me how to hold on. To Mamu, Mami, Manipal Uncle, Amrita Maasi, Karan, Honey and Kiran Didi — thank you for always being my very own personal cheerleaders. To both my dogs, Cupid and Magyk, thank you for never failing to keep the goofy child alive within me.

Optical Flow for De-Identification and Driver Behavior Classification

Author: Arjun Sarup, EECS M.S. Student, University of California, Berkeley
Faculty Advisor: Professor Gerald Friedland, Adjunct Assistant Professor EECS, UC Berkeley
Research Advisor: Dr. Meredith M. Lee, Chief Technical Advisor to the Associate Provost, UC Berkeley Computing, Data Science, and Society

Acknowledgments

A special shoutout to Professor Friedland, who made it his personal mission to shine away any difficulty that came my way. Thank you for being such an incredible mentor.
Thank you to Dr. Meredith M. Lee for teaching me what leadership, tenacity and collaboration actually means.
Thank you to Thomas Karnowski from Oak Ridge National Laboratory and Alice O'Toole from UT Dallas for always being on-call whenever I needed their help.

To the best parents in the world — my Mom and Dad. Thank you for being the Genies who made all of my wishes come true, for conjuring up the tallest trees of love and hope with your magic beans, and for making the flying carpet that will always carry me towards all of my dreams. This one, and everything that will come after this one, is for you.
All the hat-tips in the world to my Nanu, who never failed to pick me back up when all I wanted to do was stay down.
My Nani — for teaching me how to hold on, and for teaching me that those who love us never truly leave us.

To Mamu, Mami, Manipal Uncle, Amrita Maasi, Karan, Honey and Kiran Didi — thank you for always being my very own personal cheerleaders.
To both my doggos, Cupid and Magyk, thank you for never failing to keep the goofy child alive within me.
To Berkeley, for equipping me with the tools I need to embark on the adventures that come next. Go Bears!!

Abstract

In this thesis, I investigate the potential that optical flow offers for automobile driver de-identification and behavior classification. I use Farneback's algorithm for optical flow to extract the temporal movements that characterize facial behavior from a recently introduced driver-facing dataset. To enhance the behavioral variety offered within this driver-facing dataset, I introduce several data augmentation procedures that take advantage of publicly available content on media platforms like YouTube, Netflix. One of the most impactful contributions within this augmentation framework is likely our "shot-partitioner-algorithm". This shot-partitioner-algorithm utilizes human bounding boxes and changes in the cosine similarity of 2D body pose to partition videos according to the constituent camera field of views (FOVs) used in filming them. I train a classifier on the augmented data that, in its testing stage, is able to predict driver behavior for a small subset of driver-facing videos taken from the wild with a 70% testing accuracy. Finally, I carry out an experiment with 13 participants in order to qualitatively determine if visualizations of optical flow in place of the actual concerned driver-facing video can sufficiently mask driver identity.

Chapter 1: Introduction

As part of this Master's thesis, I worked with Oak Ridge National Laboratory, the West Big Data Innovation Hub based at UC Berkeley, and the MediaEval Benchmarking Initiative for Multimedia Evaluation to launch a task centered on facilitating new methods for Video Data Privacy. The following excerpt from the task description describes the motivating context that inspired my thesis work [1]:

"The lifetime odds for dying in a car crash are 1 in 107 [2]. Each year, vehicle crashes cost hundreds of billions of dollars [3]. Research shows that **driver behavior is a primary factor** in **⅔ of crashes** and a **contributing factor** in **90% of crashes** [4].

Video footage from driver-facing cameras presents a unique opportunity to study driver behavior. Indeed, in the United States, the Second Strategic Highway Research Program (SHRP2) worked with drivers across the country to collect more than 1 million hours of driver video [5, 6]. Moreover, the growth of both sensor technologies and computational capacity provides new avenues for exploration.

However, video data analysis and interpretation related to identifiable human subjects bring forward a variety of multifaceted questions and concerns, spanning privacy, security, bias, and additional implications [7]. This **task aims to advance the state-of-the-art in video de-identification**, encouraging participants from all sectors to develop and demonstrate techniques with the provided data. Successful methods balancing driver privacy with fidelity of relevant information have the potential to not only broaden researcher access to existing data, but also inform the trajectory of transportation safety research, policy, and education initiatives. [8]"

As part of this collaboration, I started analyzing the de-identification potential of optical flow, a method used to track the motion of objects within a video. I postulated that optical flow could provide a novel opportunity to anonymize the drivers within the dataset prepared by Oak Ridge National Laboratory for the Driver Video Privacy Task [1] and preserve the behavioral information provided by the dataset's subjects.

I sought to create an end-to-end process that would allow us to test this postulation. First, I used Farneback's algorithm for optical flow to extract the temporal movements that characterize facial behavior from the aforementioned driver-facing dataset.

To enhance the behavioral variety offered within this dataset, I created several data augmentation procedures that scraped publicly available content on media platforms like YouTube and Netflix. As part of the selection process, I focused on collecting data that matched or closely approximated the camera perspectives used to film the subjects in the driver-facing dataset. Additionally, in order to prioritize footage that was filmed during real-world driving conditions where subjects were in natural, unrehearsed environments, I concentrated on media content captured during a conversation, an interview, or a Mukbang [11].

The most significant contribution within the data augmentation framework is a "shot-partitioner-algorithm". This shot-partitioner-algorithm utilizes human bounding boxes and changes in the cosine similarity of 2D body pose to partition videos according to the constituent camera FOVs used in filming them.

The shot-partitioner:

1. Programmatically dissects each video selected into the camera angles that match the ones used to record the existing dataset.
2. Helps us eliminate sections of the scraped footage where the subjects are not driving or not sitting within a vehicle.
3. Creates partitions of the main footage where only the movement of the subject changes while the camera perspective remains stationary, allowing us to run optical flow on footage where only the subject's motion changes while the camera motion remains constant because the camera position does not change within any partitioned sequence.

I train a classifier on the augmented data that, in its testing stage, is able to predict driver behavior for a small subset of driver-facing videos taken from the wild with a 70% testing accuracy.

Finally, I carry out an experiment with 13 participants in order to qualitatively determine if visualizations of optical flow in place of the actual concerned driver-facing video can sufficiently mask driver identity.

Chapter 2: Related Work

Introduction to Optical Flow

Optical flow is defined as the motion that occurs between two consecutive frames of a video sequence. The motion can either be caused by the objects within the image, or the camera capturing the video. Optical flow captures this motion by tracking pixels as they move through a scene. For each pixel in an image frame, optical flow generates a displacement vector describing the x and y direction that pixel has moved from the previous frame. **Figure 1 a)** shows a ball moving in 5 consecutive frames. The arrow shows the displacement vector generated through optical flow. [12] **Figure 1 b)** shows one of the first applications of optical flow in the creation of the infamous bullet-time scene for "The Matrix".

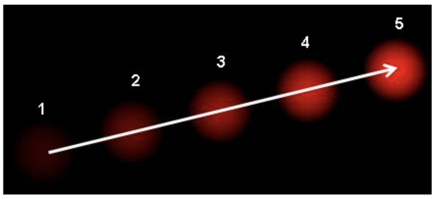


Figure 1 a): displacement vector of a ball moving in 5 consecutive frames.

Figure 1 b): For the bullet time scene in "The Matrix", the VFX team realized they didn’t have enough frames for smooth motion. They ran an optical flow pass to track where the pixels moved between adjacent frames, and created "smears" using the vectors from optical flow for interpolated, smoothened out motion [15]**.**

Optical flow assumes that:

1. An object's pixel intensity does not change between consecutive frames.
2. Pixels near each other have similar motion.

Consider a pixel (x, y) at time t with that pixel's intensity parameterized by $I(x, y, t)$. In the next frame that is captured after time $\triangle t$ passes, this pixel moves by $(\triangle x, \triangle y)$. Since the pixel itself is the same and only its location has changed while the pixel's intensity has remained constant (or so optical flow assumes), the following equation holds true:

$$I(x, y, t) = I(x + \triangle x, y + \triangle y, t + \triangle t) \qquad (1)$$

Taking the Taylor series approximation of the right hand side of (1):

$$I(x, y, t) + \frac{\delta I}{\delta x} * \triangle x + \frac{\delta I}{\delta y} * \triangle y + \frac{\delta I}{\delta t} * \triangle t \qquad (2)$$

Subtracting the left hand side in (1) from the expanded right hand side in (2):

$$\frac{\delta I}{\delta x}\triangle x + \frac{\delta I}{\delta y}\triangle y + \frac{\delta I}{\delta t}\triangle t = 0 \qquad (3)$$

Dividing (3) by $\triangle t$:

$$\frac{\delta I}{\delta x}\frac{\triangle x}{\triangle t}+\frac{\delta I}{\delta y}\frac{\triangle y}{\triangle t}+\frac{\delta I}{\delta t}\frac{\triangle t}{\triangle t}=0 \qquad (4)$$

Let $\frac{\triangle x}{\triangle t}$, or how much x moves with respect to time be u, and let $\frac{\triangle y}{\triangle t}$, or how much y moves with respect to time, be v. Then:

$$\frac{\delta I}{\delta x}u+\frac{\delta I}{\delta y}v+\frac{\delta I}{\delta t}=0 \qquad (5)$$

The other gradients in (5) are gradients related to the image's intensity, and can be readily found. However, u and v are two unknowns, and need another equation in order to be known.

Most optical flow work involves generating methods that solve u and v. These methods usually introduce an additional constraint, or an additional consideration, that helps them solve for the two unknowns. [12]

Sparse and Dense Optical Flow

Sparse optical flow approaches estimate optical flow for a limited feature set in an image. Lucas-Kanade method, a popular sparse solution, takes as input a small subset of image corners that are estimated using the Shi-Tomasi algorithm and computes optical flow for those corners. [12, 13].

Dense optical flow approaches compute displacement vectors for all the image points. A widely-used dense optical flow method is based on the polynomial expansion algorithm that Gunner Farneback presented in 2003 [12, 14].

Optical Flow for Activity Recognition

Optical Flow has been widely used within the field of Activity Recognition, oftentimes in a two-stream Convolutional Neural Network (CNN) architecture that takes in both RGB input and stacked flow fields as input. A widely-regarded two-stream solution is the Temporal Segment Networks presented by Wang et al. in [29], where they achieve 94.2% classification accuracy with RGB input and flow data on the UCF101 dataset [26]. Wang et al. used the TV-L1 algorithm [30] as their flow method of choice.

Deep Learning-based Optical Flow for Activity Recognition

There are a number of deep learning dense optical flow approaches that have increasingly been used to predict flow fields as input within action recognition networks. PWC-Net [20], FlowField [23], EpicFlow [24], Flownet [18] and Spynet [19] can be successfully used to estimate optical flow with low End-Point-Error (EPE) on ground-truth optical flow data like MPI Sintel [16] and KITTI 2015 [17]. For reference, EPE is the average euclidean distance between the ground truth and the flow estimated by a network.

Due to the dearth of ground truth optical flow data in the wild, these deep learning approaches are error-prone when faced with data that is dissimilar from what they have encountered before. When such errors are encountered for action recognition-specific tasks, corrections involve retraining the pre-trained flow networks with classification loss (how well the network is able to classify action for datasets like UCF101, HMDB [27]) instead of EPE, as described by Sevilla-Lara in [25]. The motion representation that the retrained networks now predict is similar to the flow representation, but differs in EPE at the boundary regions of human movement [25]. TV-Net in [22] also includes a similar classification loss to approximate optical flow motion representation for task-specific problems.

Optical Flow for Facial Behavior

For the purpose of this thesis, flow vectors that can capture predominantly facial motion need to be generated.

Using a deep learning-based approach to approximate optical flow could be limiting, since they are at the mercy of the dataset whose motion representation they have been trained to approximate. An example of their weak generalization capacity is seen in [28], where Allaert et al. compare pre-trained flow networks like EpicFlow, PWC-Net, and EpicFlow with classical mathematical approaches like Farneback's Optical Flow and TV-L1 within the domain of facial expression analysis. Allaert et al. train a small SVM classifier on the raw flow data predicted from each of these approaches and then compare the testing accuracy of the resulting classifiers.

Using Farneback's Optical Flow produced noticeably higher accuracy than the rest of the approaches used, since Farneback's approach was algorithmic and did not need to be pre-trained with similar data to estimate optical flow [28]. Results are summarized in **Figure 3**.

Liu et al. [31] embark on creating a two-stream CNN for driver fatigue detection — a project that closely matches this one — and use Farneback's optical flow to capture the motion representation within driver-facing videos.

After carefully considering all of these disparate options, Farneback's algorithm for optical flow remains the method of choice for extracting the motion displacement of subjects within the dataset. Besides being dataset-agnostic, Farneback's algorithm helps us avoid a potential de-identification problem inherent in most deep learning-based approaches. Networks that estimate flow can be trained on datasets to approximate facial motion as closely as possible. If such estimations are released in place of the actual data, under the assumption that optical flow is a sufficient masking procedure for the identity of the subjects who supplied that data, there is a potential security risk of including detailed identity-related data within those optical flow estimations. To visualize such phenomenon, consider **Figure 2** and the minute details included by the flow fields predicted by TV-Net when trained on the UCF101 dataset [22].

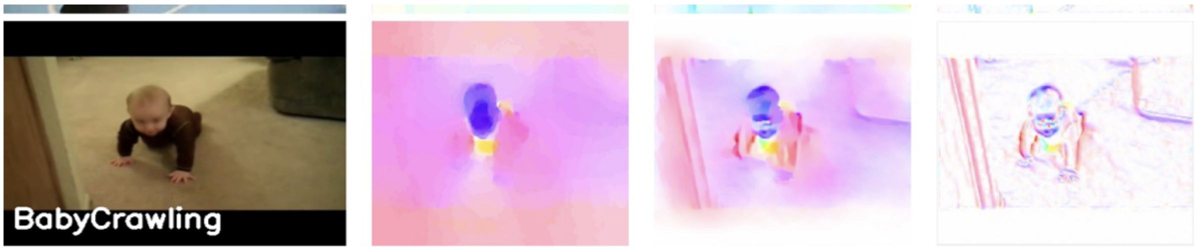


Figure 2: Left — RGB input of baby crawling. Center — TV-L1 estimated optical flow. Far Right — TVNet with Training

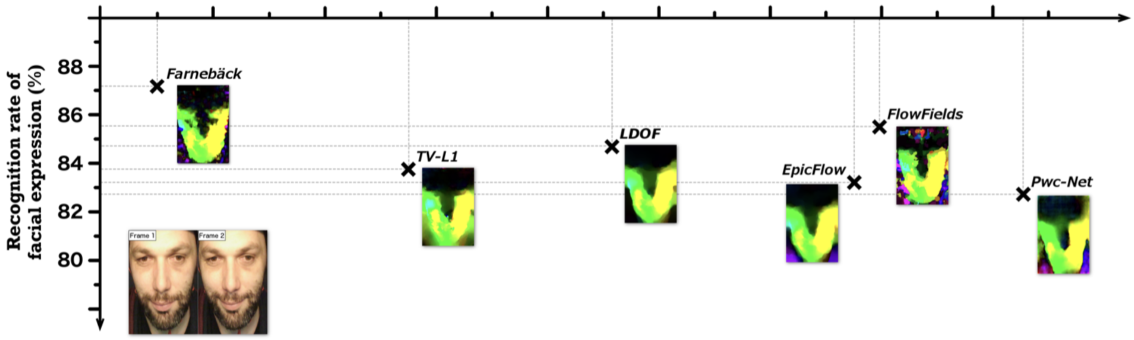


Figure 3: "Although the performance tends to be conclusive on MPI-Sintel, this is not the case for facial expression analysis, where a basic approach such as Farneback gives the best performance." [28]

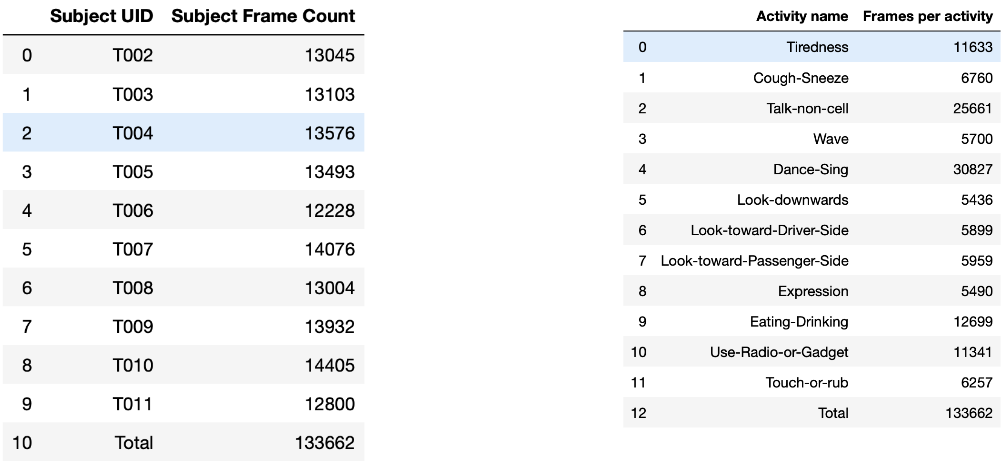
Chapter 3: Data Augmentation

Dataset Description

The ORNL dataset consists of 9 subjects who were recorded on video in a stationary Honda Civic while carrying out various activities like: **Eating/Drinking, Coughing/Sneezing, Talking, Looking downwards or to the driver's side/passenger's side**.

The videos are taken from multiple front-facing camera angles with multiple resolutions such that the same activity may have been recorded from more than one perspective. The clearest cross-section with the most visually complete information is the direct frontal view taken of the subjects in **480 by 480 colored format**. An additional sideways black-and-white field of view with **lower 360 by 240 resolution** has been provided for more data.

Refer to **Figure 4** for some key frame numbers broken down by subject and by activity type, and to **Figure 5** for visual demonstrations of each camera perspective.



	Subject UID	Subject Frame Count
0	T002	13045
1	T003	13103
2	T004	13576
3	T005	13493
4	T006	12228
5	T007	14076
6	T008	13004
7	T009	13932
8	T010	14405
9	T011	12800
10	Total	133662

	Activity name	Frames per activity
0	Tiredness	11633
1	Cough-Sneeze	6760
2	Talk-non-cell	25661
3	Wave	5700
4	Dance-Sing	30827
5	Look-downwards	5436
6	Look-toward-Driver-Side	5899
7	Look-toward-Passenger-Side	5959
8	Expression	5490
9	Eating-Drinking	12699
10	Use-Radio-or-Gadget	11341
11	Touch-or-rub	6257
12	Total	133662

Figure 4: Left — frame numbers by subject, Right — frame numbers by activity.

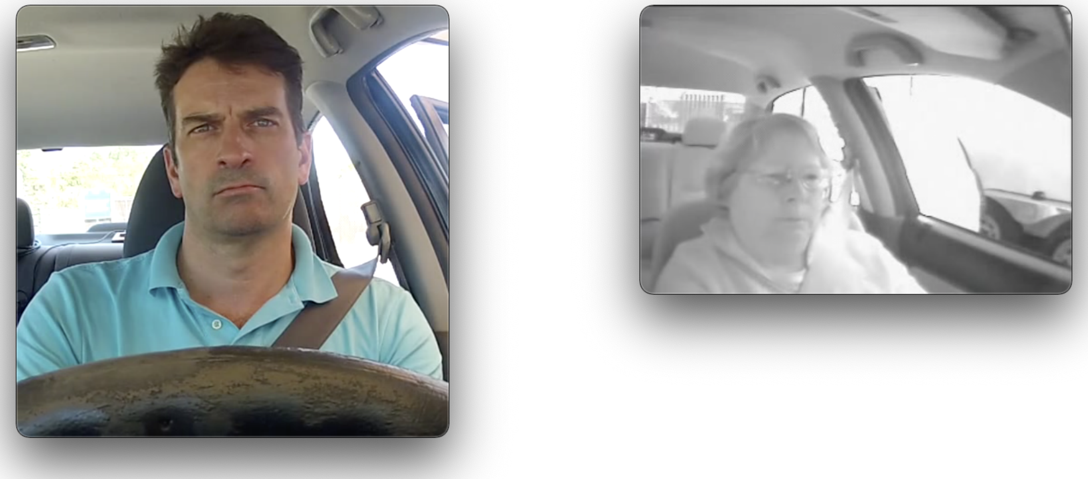


Figure 5: Left — direct frontal view of driver in $480p$ resolution. Right — side view of another driver, in lower $240p$ resolution. Both images are taken from the driver-facing dataset prepared by ORNL in [1].

Optical Flow Picks Up Background Movement

Optical flow vectors store the temporal movement of objects, surfaces, and rigid and flexible bodies as they move about in a video.

While such a feature makes optical flow a potentially useful feature source for activity prediction, the noise in motion displacement fields estimated by optical flow introduce several unwanted effects. Since optical flow is acting upon the entire pixel area of adjacent frames, it will pick up on background motion that takes place during the video. Even in the stationary Honda Civic that the ORNL dataset has been generated from, there are background movements of individuals entering the car, objects falling, or sunlight moving across the rear seats. These variable background movements are captured by the motion vectors produced by optical flow. When such vectors are fed multiple times into a classifier, they distract from the actual distinguishing features of an activity, thereby making it harder for the classifier to model and eventually predict that activity.

The dataset's activities are mostly related to and performed by the face (Tiredness, Eating/Drinking, Talking, Coughing/Sneezing). Images will ultimately be compressed to a manageable pixel resolution when fed into any classifier. If the entire image is compressed without subtracting the background and other non-activity related pixel information, potentially valuable information about motion (that could be used to disambiguate different activities) could be lost.

Facial Optical Flow allows for Background Subtraction

Liu et al. [31] avoid classification errors by removing the background and other non-facial-related information. They accomplished the background subtraction by extracting specific facial sub-features from every video, and then used the optical flow of those facial sub-features for activity prediction.

This extraction of facial features could increase the generalizability of the driver-based dataset, since there is less risk of overfitting on one particular car's interior motion and on one particular subset of facial structures.

Facial Features Extraction

Liu et. al. selected the mouth and the left eye region for extraction. For the purposes of this thesis, both the left and the right eye regions as well as the mouth were used from each video. The reason for including both the eyes is justified in the Data Augmentation section below.

The ORNL dataset ran a state-of-the-art Retina Face Detector on each video. This Face Detector, as output, generates a bounding box for the face in each frame of every video, as well as important keypoints within the face, including: the left and right eyes, the left and right corners of the lips, and the nose.

The results of the Face Detector, along with its predicted confidence, are included in csv files for every video.

I imported the csv files into dataframes, and used the following procedure to extract the mouth region for any frame i:

1. Access the left lip keypoint and the right lip keypoint from the frame.
2. Compute the euclidean distance between the two key points. Let x_1 and y_1 be the left lip keypoint's x and y coordinates respectively, and let x_2 and y_2 be the right keypoint's x and y coordinates respectively. Lip distance, or, $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
3. Treat the midpoint of the lip as the centre. The x and y midpoint coordinates can be computed through the following formulas: $y_{mid} = \frac{(y_1+y_2)}{2}$, $x_{mid} = \frac{x_1+x_2}{2}$.
4. Initialize a rectangular box with the width and height of the lip distance calculated in step 2. For opencv, the initial x and y values of this box can be generated by: $x_0 = x_{mid} - \frac{d}{2}, y_0 = y_{mid} - (\frac{d}{2})$.

For the eyes, the process described above is repeated, with the left and right irises acting as the centre and the lip distance acting as the width and the height of the rectangular box.

Figure 3 depicts the full frame, and then the mouth region and the left eye region accessed from that frame using the procedure detailed above.

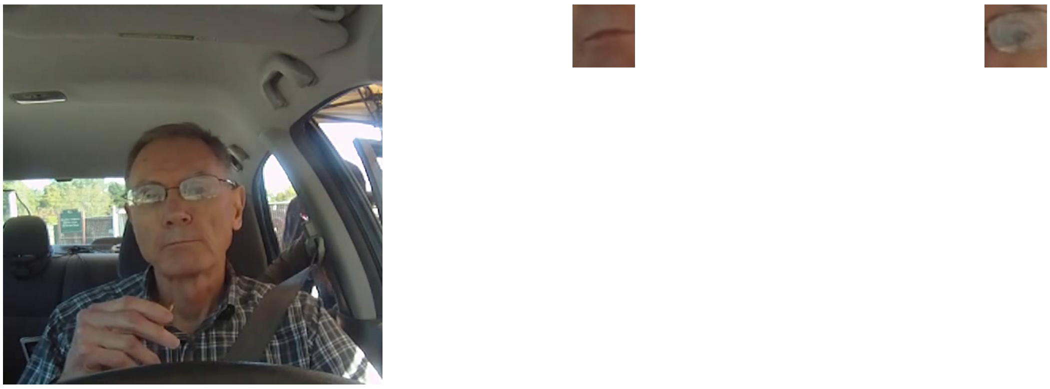


Figure 6 — Left: full frame. Center: mouth. Right: left eye. All the images in Figure 6 are taken from the driver-facing dataset prepared by ORNL in [1].

Dataset Limitations

Activities Require Manual Annotation

Most of the videos provided have segments where one or more instances of the main activity takes place interspersed with segments without notable activities. These "filler" segments can take place before the first instance of an activity, or in the interim periods between two consecutive instances.

A classifier attempting to predict an activity can potentially be thrown off by these filler segments, because the segments can have instances of another activity, or frames where there is little to no motion displacement because nothing is happening. By considering the entire video as one broad example of an activity, the classifier could potentially use the filler segments in its prediction model, which could adversely hamper classifications in-the-wild. Furthermore, the classifier may also require multiple instances of chewing or talking, essentially needing as much temporal information as possible before finally making a correct prediction.

Slicing up each video into subclips, where each clip serves as one training example of one activity, is an effective way to correct for such potential consequences. The classifier now has more examples of chewing to learn from during the training procedure instead of just one.

To slice up these videos into subclips, several activity classifiers were considered. However, most of the open-source ones are based on bodily motions and not facial motions. They do not predict the activities that are being modelled [25, 29].

Ultimately, the videos were split up into individual frames and manually annotated to label individual instances of the activity within each video and remove filler segment. The individual instances of each activity were coalesced into separate videos using ffmpeg, a popular library generally utilized for frame analysis and image compression.

Driver-Facing Data does not have enough examples of Environmental and Facial Variation

In **Figure 6**, notice that the subject is wearing spectacles, causing the spectacles to reflect some of the outside environment. For activities like "Tiredness" or "Coughing/Sneezing" that depend on eye movement, motion displacement fields calculated for the eye region will not only include eye movement, but also movement from the outside environment (since that outside environment will be picked up by subjects wearing spectacles).

This observation of the data isn't a bug, but a feature. In the real-world, despite isolating the facial features, the outside environment will appear in some form within the motion data captured by optical flow. Light can stream across a subject's face or, depending on the time of the day, be so bright that certain facial features are completely occluded. This is exactly the case in Fig. 4, where light from the Sun or from the adjacent environment has blocked off necessary pixel information for the mouth and the eyes.

All of these real-world issues are necessary to include for any classifier to be robust to the noise and chaos of driving around in a non-stationary, dynamic environment. However, there is a dearth of training examples to accurately account for such phenomena. The subject whose face is occluded by light in **Figure 7** only has about ~1000 frames of data across all activities, and dividing up those frames into specific instances of those activities only leads to around 1 or 2 instances per activity where the lighting issue takes place.

Other facial features can vary, like lip shape, the contours and the hair (or lack thereof) around the mouth, the wrinkles around the eyes. These features, for activities like Chewing or Eating or Sneezing, will have distinct motion fields for different people. Additionally, motion fields can also vary across different perspectives. In the real world, if somebody has positioned their camera at a sideways angle or an angle that hasn't been covered by the existing dataset, the accuracy of the classifier may go down if it hasn't learned from similar angles in its training stage.

Even behavior for the same activity can vary. Some people may like to talk or chew or sing with their mouth significantly open.

For all these considerations, the quantity of training data given, while acting as an effective exposure and initial introduction to the kind of variation any dataset should have, isn't enough to create a robust classifier that will be able to predict driver activity/behavior using temporal optical flow information.

For any optical flow approach to be demonstrably effective, there needs to be a dataset with enough variety in its training data. Without extensive examples to capture facial and environmental variation, it becomes necessary to develop data augmentation approaches that will diversify the data with more training examples.

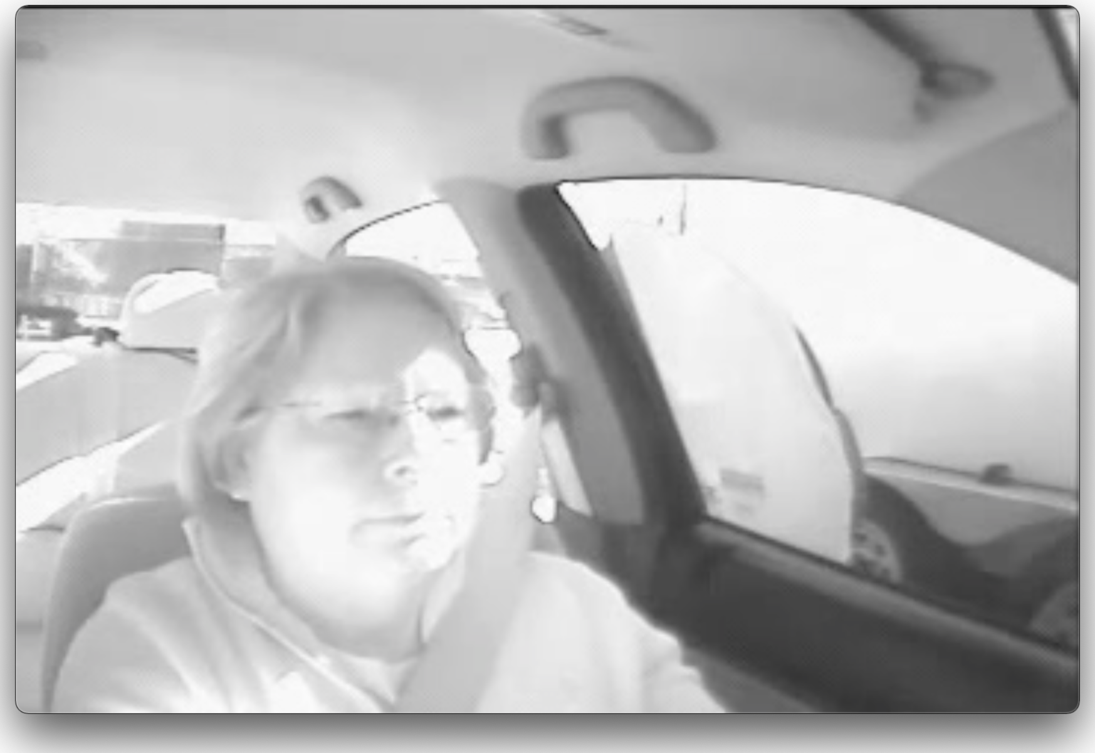


Figure 7: Subject with face occluded by light. The image is taken from the driver-facing dataset prepared by ORNL in [1].

Manual Data Augmentation

Open-Source Activity Recognition Data does not Contain Examples of Driver Behavior

Datasets like the widely used UCF101 [26], Kinetics-400 from Deep Mind could be used for training if the solution under development was for activities related to bodily motion. However, these datasets do not have any facial-motion related examples.

The National Tsing Hua University-Driver Drowsiness Detection Dataset [32] is a similar dataset in that it attempts to capture different types of driver behavior, but the footage recorded is not inside an actual vehicle. The footage is captured against a static background and the camera perspectives used are not similar to the camera angles that would be possible within a vehicle.

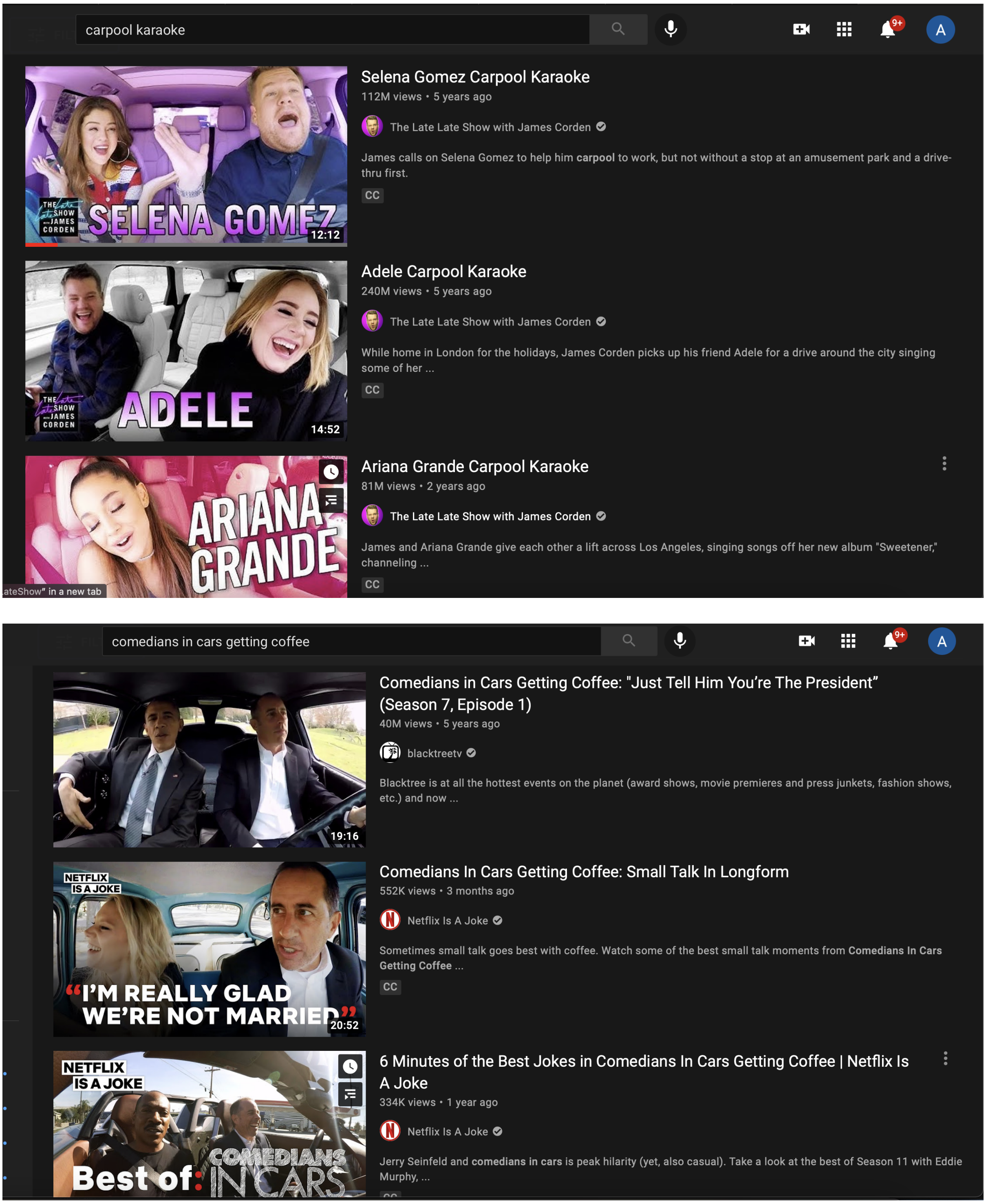
Faced with the lack of open-source labelled data, the only viable alternatives seemed to be recording limited datasets myself and manually looking for publicly available footage through YouTube, Netflix and similar content-oriented platforms.

In-Car Conversational Footage Provides Resource for Data Augmentation

At first glance, car reviewing channels on YouTube could function as a good source for data augmentation. After a thorough vetting of the general footage they capture, I observed that the reviewing channels' focus on showcasing the car's features instead of giving a clear look at the driver and their predominant use of cinematic, artificial lighting makes them unsuitable for this project.

After digging a bit further, programs like "Carpool Karaoke" and "Comedians in Cars Getting Coffee" provide a good match for the kinds of effective training data that I was looking to access. While the former is a segment where popular television host James Corden drives around with various musicians as they talk, sing together, and get food inside a car, the latter features celebrated comedian Jerry Seinfeld as he showcases a mixture of vintage and modern cars by driving in them to get coffee with another celebrity.

Popular YouTube downloader extensions can be used to scrape the "Carpool Karaoke" playlist from YouTube, while the videos for "Comedians in Cars Getting Coffee" will require a Netflix subscription, Google Chrome, and a PC capable of screen recording.



carpool karaoke

Selena Gomez Carpool Karaoke
112M views • 5 years ago
The Late Late Show with James Corden
James calls on Selena Gomez to help him **carpool** to work, but not without a stop at an amusement park and a drive-thru first.
CC

Adele Carpool Karaoke
240M views • 5 years ago
The Late Late Show with James Corden
While home in London for the holidays, James Corden picks up his friend Adele for a drive around the city singing some of her ...
CC

Ariana Grande Carpool Karaoke
81M views • 2 years ago
The Late Late Show with James Corden
James and Ariana Grande give each other a lift across Los Angeles, singing songs off her new album "Sweetener," channeling ...
CC

comedians in cars getting coffee

Comedians in Cars Getting Coffee: "Just Tell Him You're The President" (Season 7, Episode 1)
40M views • 5 years ago
blacktreetv
Blacktree is at all the hottest events on the planet (award shows, movie premieres and press junkets, fashion shows, etc.) and now ...

Comedians In Cars Getting Coffee: Small Talk In Longform
552K views • 3 months ago
Netflix Is A Joke
Sometimes small talk goes best with coffee. Watch some of the best small talk moments from **Comedians In Cars Getting Coffee** ...
CC

6 Minutes of the Best Jokes in Comedians In Cars Getting Coffee | Netflix Is A Joke
334K views • 1 year ago
Netflix Is A Joke
Jerry Seinfeld and **comedians in cars** is peak hilarity (yet, also casual). Take a look at the best of Season 11 with Eddie Murphy, ...

Figure 8: Samples of the Driver-Facing Footage from "Carpool Karaoke" and "Comedians in Cars Getting Coffee"

Carving out Relevant Footage by Hand is Time Consuming

I started off by trying to manually slice out different clips from these individually long 12-15 minute episodes. The process was extremely onerous, since I had to first search for subsections in each episode where the subjects were actually in a vehicle, and then manually carve out frame numbers within the filtered footage where the camera was stationary.

Optical Flow Models Camera Motion

In both "Comedians in Cars Getting Coffee" and "Carpool Karaoke", there are multiple wide shots and closeup shots of the vehicle's subjects.

Optical Flow picks up on all the motion that takes place during a video, and that can be motion related to the camera. Therefore, if there is minimal change in the movement of a subject between adjacent frames, but a significant shift in camera perspective, optical flow will approximate that motion. Since I was searching for training examples where the subject is moving, I needed a method to slice out subsections of footage where the camera position remains constant.

Shot Partitioner Programmatically Slices Essential Data

I developed a customized algorithm that utilized bounding boxes, 2D body pose, and change in cosine similarity to programmatically slice out the essential, driver-focused data from the videos I scraped.

Assume a video V has been recorded through two wide shots. Using the "shot-partitioner" algorithm, I partition V shot-wise, such that all the driving segments $A_1, A_2, ...A_k$ corresponding to wide shot type A wind up in one collated clip C_1 and all the driving segments $B_1, B_2, ...B_k$ corresponding to wide shot type B wind up in another collated clip C_2. This procedure helps us get shot-specific data. Without this partition, flow motion estimated from adjacent frames can appear to show the participants moving significantly when in fact, only the camera angle changed.

The shot-partitioner has the following components:

1. A sitting detector based on heuristics from estimated 2D Body Pose to predict if the subjects are sitting or not. This functions as an initial filter to slice out sections of the footage where the subjects are sitting in a vehicle.
2. A cosine similarity test that disambiguates between different shots.

```
def shot_partitoner(frame_curr, frame_prev):
  detectron_data = # Run Detectron on frame_curr here #
  num_human_bounding_boxes = # Extract num_bounding_boxes from detectron_data.
  if 1 <= num_human_bounding_boxes <= 2:
      body_pose_x, body_pose_y = (
        # body_pose_x has the x coordinates of all the
        # body parts, body_pose_y has the y coordinates of
        # all the body parts
      )
      sit = are_they_sitting(body_pose_x, body_pose_y)
      if sit:
        if num_human_bounding_boxes == 2:
          num_change = # get previous frame's bounding box number, subtract
          # from num_human_bounding_boxes.
          if num_change > 0:
            # if Num change has changed from 1 to 2, frame has
            # switched Identify if this
            # shot is new by doing nearest neighbor analysis with other
            # previously seen wide shots.
          else:
            # If bounding box number hasn't changed wrt previous frame, then the
            # following block will execute.
            person1_index, person2_index = (
              # Let person 1 be the one with the lower x coordinate,
              # let person 2 be the one with the higher x coordinate
            )
            person1_cosine_similarity_x, person2_cosine_similarity_x = (
              # Perform cosine similarity test for x coordinates with
              # previous frame here
            )
            person1_cosine_similarity_y, person2_cosine_similarity_y = (
              # Perform cosine similarity test for y coordinates with
              # previous frame here
            )
            pass_similarity_test = (
              # False if any of the cosine similarities are more than 30 degrees
              # True otherwise
            )
            if pass_similarity_test:
              .# Add current frame to previous frame's shot type.
            else:
              # add new shotType to list of shots already tracked.
              # Append current frame to shotType.

        if num_human_bounding_boxes == 1:
          # similar procedure as above, just with one person in the current frame
          # whose joint similarity should be compared with the previous frame's
          # if the previous frame also only has 1 bounding box.

def are_they_sitting(body_pose_x, body_pose_y):
  # Peform sitting test #
```

Sitting Detector

I design this heuristic by first assuming that whenever the subjects sit down for the first time, they sit down within the car. My second assumption is that, for any frame where somebody is sitting down, the confidence score for their lower body part 2d keypoints will be very low, since those lower limbs will not be visible. These assumptions should only be made for this specific search space of videos.

The algorithm first checks to see if frame i has at least 1 and not more than 2 bounding boxes with confidence scores more than the high 90s (if it has more than two, then that means they have already stepped foot into the restaurant to have coffee and the video does not have any more driver-facing footage).

I use the COCO Keypoint RCNN model from Detectron2, a widely used open-source state-of-the-art computer vision library [33], to extract the bounding boxes of all the humans in frame i (intialized as frame i in the code block) of video V. This can be done by following the instructions in the sample Jupyter Notebook linked here. A visual illustration of Detectron's output can be found in Figure 9 below.

I access the 2D body pose keypoints returned by the model on frame i. These body pose keypoints contain the (x, y) location within the current 2D frame i of 17 different body joints. Certain body parts, like lower limbs and feet could be hidden from view since the person is sitting down for the duration of "Carpool Karaoke".

Low confidence scores between 0.1-0.3 for the lower limbs would imply difficulty in being able to predict the lower limbs' location, and this difficulty would further imply that the two subjects are sitting down. I can use the tangent line test to figure out if the upper leg and the lower leg form an angle that is less than or equal to 90 degrees (for sitting postures, the upper leg and the lower leg below the knee is either perpendicular or less than perpendicular).

Cosine Similarity Test

Based on the three types of wide shots these videos have, I assume that the relative position of the two subjects within the car won't change. A passenger in the front seat will always occupy the left side of the frame, while a driver in the front seat will always occupy the right side of the frame. **Figure 9** demonstrates this phenomenon. If the shot under consideration is a wide (aka, there is more than one person), I associate the bounding boxes with the concerned people by looking at the x coordinate of the boxes. Smaller x will belong to the person to the left, and larger x for the person to the right. This association will help getting the index to access the keypoints of the person on the left, and the keypoints of the person on the right.

The rest of the process summarized in the code block relies on cosine similarity. If $v_i x$ and $v_i y$ are the x and y vectors of the body parts in frame i respectively, $v_{i-1} x$ and $v_{i-1} y$ are the x and y vectors of the body parts in frame $i - 1$ respectively, then the cosine similarity is computed by the below equation:

$$\frac{v_{i-1}j * v_i j}{||v_{i-1}j|| * ||v_i j||} \text{ for } j \in x, y$$

If the vectors are dissimilar for any one person in the car (in practice, this means if their angle similarity is more than 30 degrees --- an arbitrary threshold that works well here), then the camera switched to a different view in frame i.

The nearest neighbor analysis described in the code block involves doing a cosine similarity test with previously encountered shot types that have the same number of bounding boxes as the current frame. This similarity test is done with an average of the last 5 frames' 2D body parts that

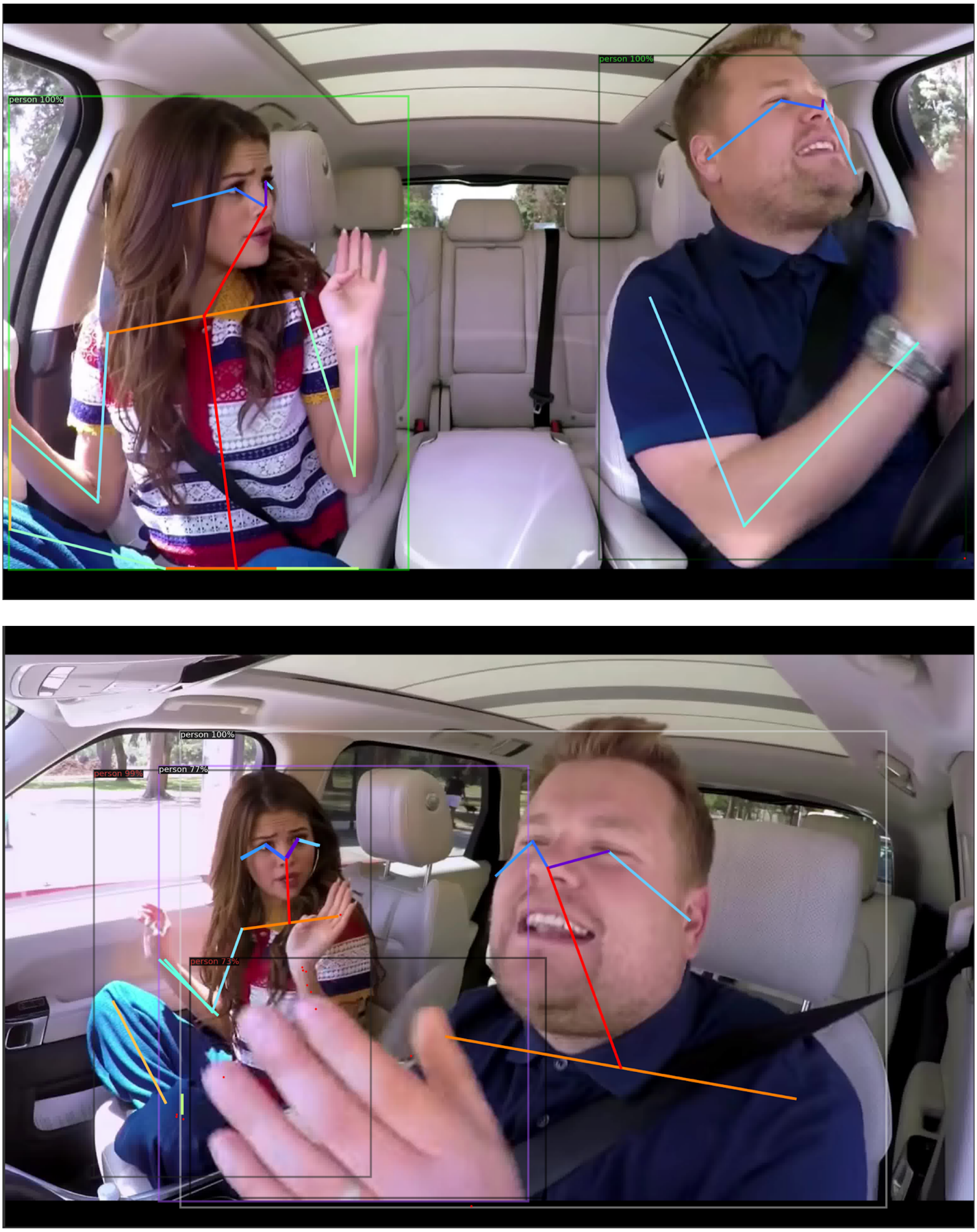
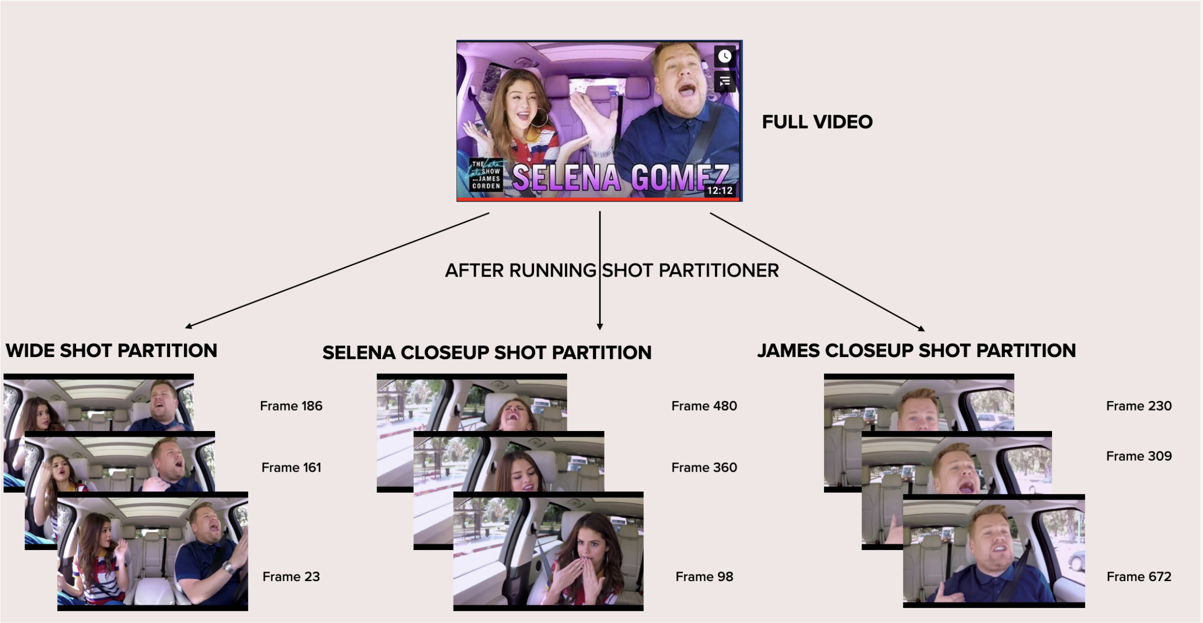


Figure 9: Results of Running Detectron2 on two Adjacent Frames from "Carpool Karaoke"

Illustration of Shot Partitioner Algorithm



Manually Annotate Resulting Partitions

After this raw data is collected from each video, I sift through all the clips extracted, remove filler segments and frames included erroneously, and manually label/split off the instances that correspond to activities in the ORNL dataset into further subclips. For example, if a closeup collated shot video has multiple instances of talking and expressions, each of those instances for that closeup shot video is separated into a different clip to provide as many training references for different activities to the classifier as possible.

This procedure is still time-consuming, but the more arduous manual filtering work of first slicing out the sub-clips and then collating them shot-wise is taken care of by the heuristics described above.

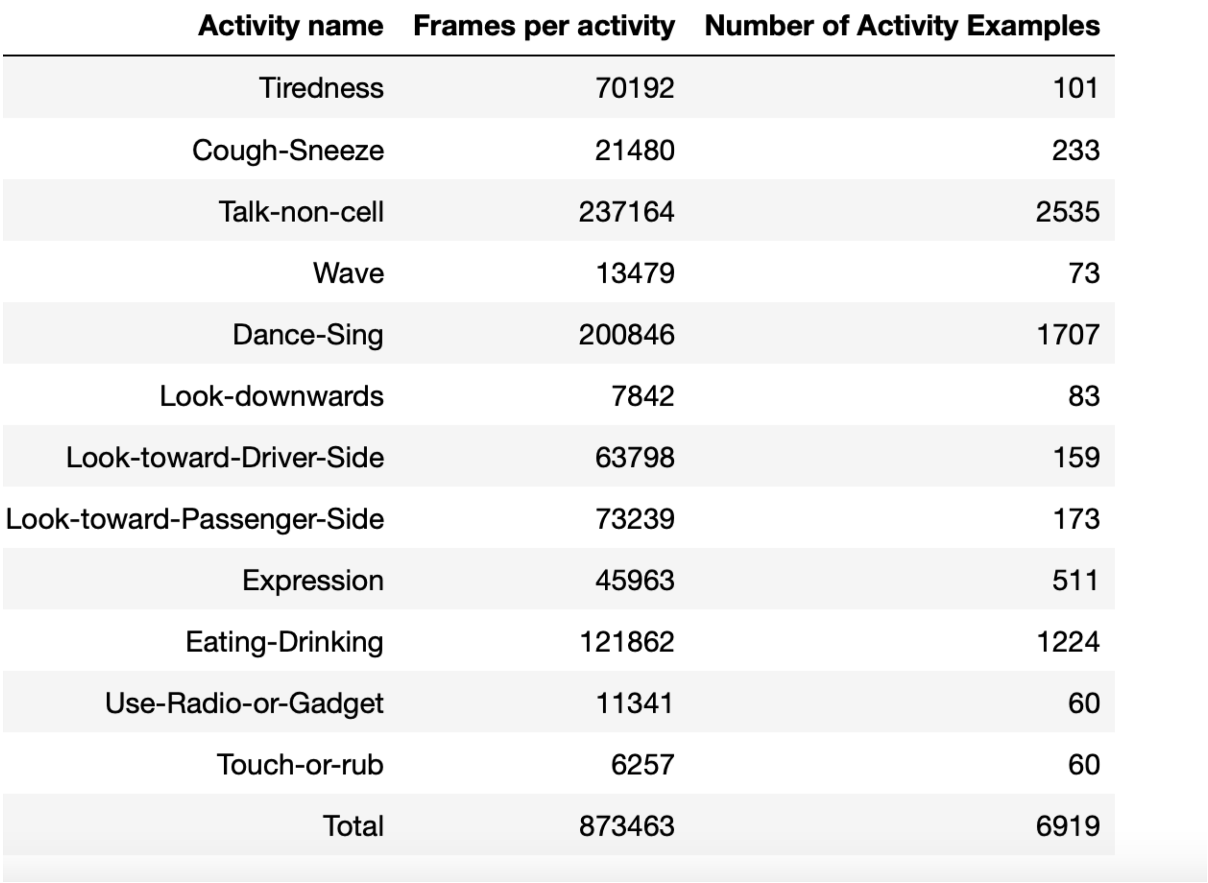
Used Both Passenger and Driver Behavior in Data Augmentation

Both Passenger and Driver Behavior was used in Data Augmentation because there weren't distinguishing characteristics between the two. Besides who was driving the car, the subjects being recorded in the video clips that are added have similar lighting, similar camera perspectives, and, above all, offer different examples of the same behavior. Therefore, it was logical to consider both subjects in the front-seat.

Since these subjects look and interact with both sides of the car (driver's side and passenger's side), I used the right and left eye vectors in my classifier, instead of just the left eye region as postulated by Liu et al.

Augmentation Procedure Produces Wider Variety of Training Examples

With "Comedians in Cars Getting Coffee" and "Carpool Karaoke", there is enough variety in training examples, subjects, lighting and camera perspectives. "Carpool Karaoke" helps beef up the number of training examples for "Eating/Drinking", "Singing/Dancing", and "Talking" significantly. "Comedians in Cars Getting Coffee", on the other hand, utilizes so many different cars that it is able to effectively provide a lot of examples that have differences in the same wide/closeup perspective. Since "Comedians" also undertakes longer journeys, the movement of light is more effectively captured. Consider **Figure 11**, where light streams across Jerry's face as he talks, thereby adding to some of the light source movement captured by the initial ORNL dataset. Additionally, "Comedians" helps us include more examples of subjects looking downwards (when they reach for their seatbelts), looking sideways (when they talk to each other), and various expressions. The frame counts and the number of examples per activity after the inclusion of this data is included in the table depicted in **Figure 10.**



Activity name	Frames per activity	Number of Activity Examples
Tiredness	70192	101
Cough-Sneeze	21480	233
Talk-non-cell	237164	2535
Wave	13479	73
Dance-Sing	200846	1707
Look-downwards	7842	83
Look-toward-Driver-Side	63798	159
Look-toward-Passenger-Side	73239	173
Expression	45963	511
Eating-Drinking	121862	1224
Use-Radio-or-Gadget	11341	60
Touch-or-rub	6257	60
Total	873463	6919

Figure 10: Dataset Details After Augmentation Procedure

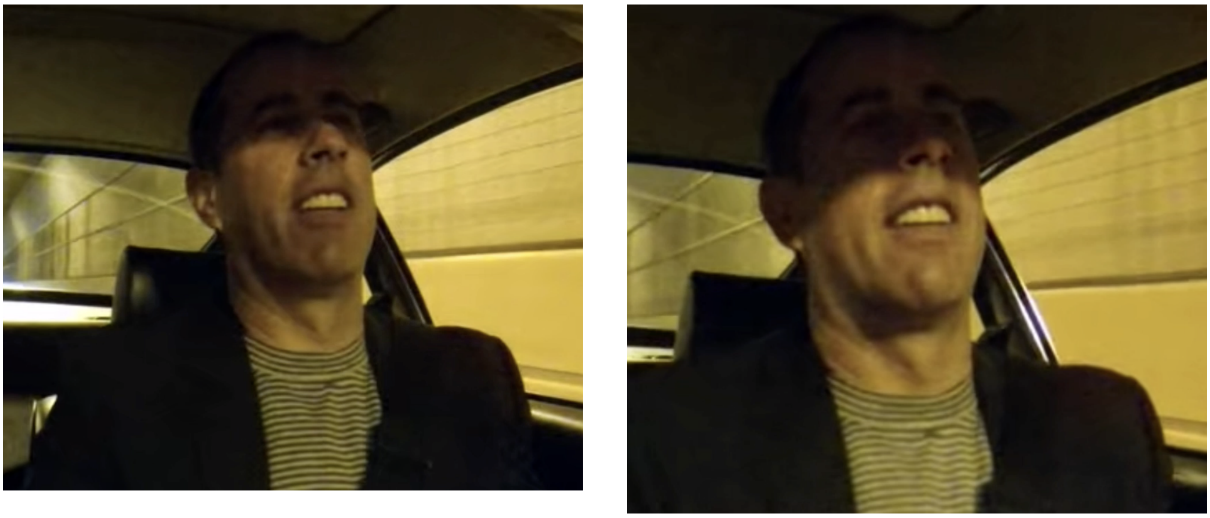


Figure 11: Light Streams Down Jerry's Face in two adjacent frames from "Comedians in Cars Getting Coffee"

Modified Apple's Vision API To Identify Facial Landmarks for Pre-Recorded Footage

The data collected through this augmentation procedure still has to be filtered so that the individual facial features can be extracted from every new clip. Google Cloud's Vision API is cost-prohibitive to researchers, and running a custom detector on all the ~700,000 new frames could require resources not readily available.

Apple's Vision API includes a sample app, "Tracking the User's Face in Real Time", that can be modified to run on iPhones for pre-recorded videos. While this solution limits its adaptiveness to users of iPhones, the strength of the tracking offered and the similarity in the facial features that are tracked offered a viable option for this project.

The procedure for isolating the mouth and the eyes is covered in the "Facial Features Extraction" section of Chapter 3.

Augmented the "Tiredness" Activity through Personally Recorded Footage

On average, there are more than 300 training examples for each activity except for the "Tiredness" activity. It is challenging to find publicly available driver-facing data out in the wild where people are yawning, blinking slowly or dilating their eyes — all examples of "Tiredness".

Eventually, I personally recorded approximately ~43,000 frames for "Tiredness", increasing the number of training examples to 109. The recording was done in a stationary Hyundai Tucson with a front-facing camera attached to a gimbal mount. About 27,000 frames came from the lead researcher of this paper, and the rest came from people in his neighborhood and family.

Chapter 4: Model Architecture and Testing

The model classifier ultimately selected out of SVM, Random Forest, and deep learning approaches was the deep learning approach, since there was extensive research that a specific two-stream CNN produced state-of-the-art results on widely known action recognition datasets.

The architecture of the CNN selected is detailed below. It is a temporal segment network with two separate temporal ConvNet streams. One ConvNet takes in the temporal data of the mouth region, while the other takes the temporal data of the eye region.

Besides replacing the spatial ConvNet in the two-stream TSN with a temporal ConvNet, the rest of the model creation and model training procedure is similar to the action recognition-specific approach presented by Wang et. al [29]. Their two-stream ConvNets received state-of-the-art results on popular datasets like UCF101, which is why I looked at their network architecture and problem space as the closest problem that was similar to my own.

"Formally, given a video V, we divide it into K segments of equal durations. Then, the temporal segment network a sequence of snippets as follows: $TSN(T_1, T_2, ..., T_k) = H(G(F(T_1; W), F(T_2; W), F(T_k; W)))$

Here (T_1, T_2, T_k) is a sequence of snippets. Each snippet T_K is randomly sampled from its corresponding segment S_k. $F(T_k; W)$ is the function representing a ConvNet with parameters W which operates on the short snippet T_k and produces class scores for all the scores. The segmental consensus function, G, evenly averages the outputs from multiple short snippets to obtain a consensus of class hypothesis among them. Here, we choose the widely used Softmax function for H. The initial loss function is a standard cross-entropy loss and is formed as:

$$\mathcal{L}(y, \mathbf{G}) = -\sum_{i=1}^{C} y_i \left(G_i - \log \sum_{j=1}^{C} \exp G_j \right),$$

Where C is the number of action classes and y_i is the groundtruth label concerning class i." [29] The number of snippets is set to 6 according.

Each snippet T_k contains optical flow data of 5 frames, where each frame is treated as a separate channel. Therefore, if the dimensions of the mouth region are 50*50 for every training example, and the number of channels for every snippet is 5 frames, then the input into the network is an optical flow network with shape (50, 50, 5).

"We use the mini-batch stochastic gradient descent algorithm to learn the network parameters, where the batch size is set to 256 video clips and momentum set to 0.9. For temporal networks, we initialize the learning rate as 0.005, which reduces to its 1/5th after 12000 and 18,000 iterations. The maximum iteration is set as 20,000." [29]

The Training Accuracy is 72.17%, and testing accuracy on the 50 clips pulled for each of the activity types from subjects not seen in the augmented dataset is 68.33%. These clips were pulled from the popular Indian conversational YouTube show:

"The Bombay Journey", which has similar driver-facing behavior as the main dataset.

"Touch-or-Rub"'s testing and "Use-Radio-or-Gadget"'s testing was done by holding out subject T002's and T011's video clips in the training procedure.

Chapter 5: Exploring Optical Flow's De-Identification Potential and Next Steps

Study to Test Optical Flow's De-Identification Potential

To preliminarily test the de-identification potential of optical flow vectors, I picked 13 participants and asked them to identify subjects based on their optical flow data. I saved optical flow data for the entire image frame as hsv instead of separate flow fields. HSV also allowed me to test the best possible visualization of optical flow, as the grayscale-based flow field visualization does not reveal as much visually.

I carried out two tests for each gender: male and female. Each participant would be presented a series of optical flow clips related to a specific activity performed by one subject. These clips are from the more easily identifiable activities, i.e., signing and talking. After this presentation, they are given 5 possible options for the identity of the subject, one of whom is the actual identity for the subject. They can either:

1. choose from the 5 possible options,
2. forfeit by admitting that they cannot guess, or,
3. ask to see more clips.

If they choose option 3, the loop repeats.

Should they choose to pick option 1, I asked one followup question before revealing the result of their guess: was their choice random or did they pick up on specific features within the optical flow clips that correlated with the subject’s identity? This followup question was asked before revealing the correct guess so that the result does not bias their initial perception of the clips they viewed.

I got clips of Selena Gomez from “Carpool Karaoke” for the female test. I showed each participant pictures of Gomez, Eilish, Cardi B, Ariana Grande, Taylor Swift after showing them the initial clips. These pictures functioned as one of the five available options they could pick for the subject within the videos.

I carried out another test that had Seth Meyers’s clips from his “Comedians in Cars Getting Coffee” episode. I reduced the space of possible guesses to Meyers, John Mulaney, Bo Burnham, Matt Bomer, and Hrithik Roshan.

For the female test, 2 participants were able to correctly identify the subject, 8 admitted that they didn’t know, and 3 picked an incorrect guess. Nobody chose to see additional clips before making their guess. Out of the 5 who chose to make a guess, before being told the outcome of their guess, both admitted that they randomly chose, and that all of the possibilities were equally likely.

For the male test, 3 participants asked to see more clips before making incorrect guesses, while 9 admitted that they didn’t know what the answer was. Again, the participants who picked the incorrect guesses, before being told the outcome of the guess, admitted that they chose randomly.

Obviously, the experiment is limited. None of the participants had access to compute power that they could use to de-identify the subject they saw. They also did not have the benefit of working together to identify the subjects.

However, based on the data collected, optical flow, from a purely qualitative standpoint, does have potential as a de-identification approach.

Next Steps

The purpose of this thesis was to act as an initial proof of concept that could demonstrate the efficacy of optical flow as a de-identification procedure and as an approach for preserving the motion activity characterizing facial behavior.

Most of this thesis was focused on creating a data augmentation framework that could increase the generalizable capacity of behavioral classifiers trained on optical flow when they are tested on videos taken from the wild. The highlight of working on this augmentation framework was creating the camera shot partitioner algorithm that I used to automatically extract relevant information from the multitude of videos scraped off the web.

The end-to-end procedures I created can act as a springboard to launch further research efforts into the distinguishable motion characteristics that optical flow encodes for different kinds of driving behavior. Further concrete experimental studies could also help demonstrate and evaluate how useful optical flow could be in supporting data privacy while enabling valuable behavioral analysis.

Bibliography

[1] *Driving Road Safety Forward: Video Data Privacy.* Retrieved August 10, 2021, from https://multimediaeval.github.io/editions/2021/tasks/videoprivacy/

[2] *Odds of dying*. (2021, March 04). Retrieved March 28, 2021, from

https://injuryfacts.nsc.org/all-injuries/preventable-death-overview/odds-of-dying/

[3] Blincoe, L. J., Miller, T. R., Zaloshnja, E., & Lawrence, B. A. (2015, May). *The economic*

and societal impact of motor vehicle crashes, 2010. (Revised)(Report No. DOT HS 812 013).

Washington, DC: National Highway TrafficSafety Administration.

[4] Dingus, T., Guo, F., Lee, S., Antin, J., Perez, M., Buchanan-King, M., & Hankey, J. (2016, March 08). *Driver crash risk factors and prevalence evaluation using naturalistic driving data*. Retrieved April 18, 2021, from https://www.pnas.org/content/113/10/2636

[5] *About Safety Data: Strategic Highway Research Program 2* (SHRP 2). Retrieved from http://www.trb.org/StrategicHighwayResearchProgram2SHRP2/SHRP2DataSafetyAbout.aspx

[6] *A Brief Look at the History of SHRP2.* http://shrp2.transportation.org/pages/History-of-SHRP2.aspx

[7] Finch, K. (2016, April 25). *A visual guide to practical data de-identification*. Retrieved March 28, 2021, from https://fpf.org/blog/a-visual-guide-to-practical-data-de-identification/

[8] Exploratory Advanced Research Program Video Analytics Research Projects https://www.fhwa.dot.gov/publications/research/ear/15025/15025.pdf

[9] Ferrell, R., Aykac, D., Karnowski, T., & Srinivas, N. (2021, January). *A Publicly Available, Annotated Data Set for Naturalistic Driving Study and Computer Vision Algorithm Development*. Retrieved from https://info.ornl.gov/sites/publications/Files/Pub122418.pdf

[10] Baragchizadeh, Asal, O'Toole, Alice, Karnowski, Thomas Paul, & Bolme, David S. *Evaluation of Automated Identity Masking Method (AIM) in Naturalistic Driving Study (NDS)*. United States. https://doi.org/10.1109/FG.2017.54

[11] "Mukbang." Wikipedia, Wikimedia Foundation, 10 Aug. 2021, en.wikipedia.org/wiki/Mukbang.

[12] "Optical Flow." OpenCV, docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html.

[13] Baker, Simon, and Iain Matthews. "Lucas-Kanade 20 Years on: A Unifying Framework." International Journal of Computer Vision, Kluwer Academic Publishers, link.springer.com/article/10.1023/B:VISI.0000011205.11775.fd.

[14] Farnebäck, Gunnar. "[PDF] Two-Frame Motion Estimation Based on Polynomial Expansion: Semantic Scholar." www.semanticscholar.org/paper/Two-Frame-Motion-Estimation-Based-on-Polynomial-Farnebäck/534805683c27accb27d66d9425f759b798df380a.

[15] https://www.youtube.com/watch?v=1VxWKmnkuI0

[16] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, European Conf. on Computer Vision (ECCV), Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012

[17] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. In ISPRS Workshop on Image Sequence Analysis (ISA), 2015.1.

[18] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.

[19] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol- ume 2, page 2. IEEE, 2017.

[20] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[21] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[22] Lijie Fan, Wenbing Huang, Stefano Ermon Chuang Gan, Boqing Gong, and Junzhou Huang. End-to-end learning of motion representation for video understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6016–6025, 2018.

[23] C. Bailer, B. Taetz, and D. Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement opical flow estimation. In International Conference on Computer Vision (ICCV), 2015.

[24] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. EpicFlow: Edge-Preserving Interpolation of Correspon/dences for Optical Flow. In Computer Vision and Pattern Recognition (CVPR), 2015.

[25] Sevilla-Lara, L., Liao, Y., Güney, F., Jampani, V., Geiger, A., & Black, M. J. (2018, October). On the integration of optical flow and action recognition. In German Conference on Pattern Recognition (pp. 281-297). Springer, Cham.

[26] K. Soomro, A. Roshan Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. In CRCV-TR-12-01, 2012.

[27] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In International Conference on Computer Vision (ICCV), 2011.

[28] Allaert, B., Ward, I. R., Bilasco, I. M., Djeraba, C., & Bennamoun, M. (2019). Optical Flow Techniques for Facial Expression Analysis--a Practical Evaluation Study. arXiv preprint arXiv:1904.11592.

[29] Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., & Van Gool, L. (2016, October). Temporal segment networks: Towards good practices for deep action recognition. In European conference on computer vision (pp. 20-36). Springer, Cham.

[30] Sánchez, Javier & Meinhardt-Llopis, Enric & Facciolo, Gabriele. (2013). TV-L1 optical flow estimation. Image Processing On Line. 3. 137-150. 10.5201/ipol.2013.26.

[31] Liu, Weihuang & Qian, Jinhao & Yao, Zengwei & Jiao, Xintao & Pan, Jiahui. (2019). Convolutional Two-Stream Network Using Multi-Facial Feature Fusion for Driver Fatigue Detection. Future Internet. 11. 115. 10.3390/fi11050115.

[32] Ching-Hua Weng, Ying-Hsiu Lai, Shang-Hong Lai, “Driver Drowsiness Detection via a Hierarchical Temporal Deep Belief Network”, In Asian Conference on Computer Vision Workshop on Driver Drowsiness Detection from Video, Taipei, Taiwan, Nov. 201

[33] Yuxin Wu, Alexander Kirillov, Francisco Massa and Wan-Yen Lo, & Ross Girshick. (2019). Detectron2. https://github.com/facebookresearch/detectron2.