2021

Estimating Heading from Optic Flow with Neural Networks

Natalie T. Maus
Colby College

Computer Science Honors Thesis Project 2020-2021

Estimating Heading from Optic Flow

By: Natalie Maus
Thesis Advisor: Professor Oliver Layton

Abstract:

Humans have a remarkable ability to estimate their direction of self-motion, or heading, based on visual input stimulus (optic flow). Machines, on the other hand, have a difficult time with this task, especially when flow is introduced that is inconsistent with the motion of the observer. For example, when moving objects enter the field of view, their motion provides inconsistent flow data which often disrupts heading estimates of current heading estimation models. We investigate the ability of neural networks to estimate heading from optic flow data and the limitations of these models when different variations of inconsistent flow are introduced. Since convolutional neural networks (CNNs), in particular, have been shown to be ideal for image and video analysis tasks, we focus our exploration on neural networks with convolutional architectures. We show that a simple CNN can be trained to effectively estimate heading from optic flow image data with human-like accuracy. We further investigate the advantages of using a convolutional recurrent neural network (RNN) as opposed to a simple CNN to estimate heading. Our findings suggest that the ability of an RNN to take into account the temporal dimension provides an advantage when it comes to several types of inconsistent flow data. These advantages may justify the added complexity and more costly training time of an RNN versus a CNN when it comes to estimating heading from Optic Flow.

Introduction:

The term 'optic flow' is commonly used to describe the pattern of apparent motion of objects and surfaces in a visual scene caused by the relative motion of the observer. More precisely, optic flow is the displacement of points on the eye of the observer while the observer is moving through the world. For example, when you as the observer are driving a car and you look out the front windshield, the world outside appears to be moving towards you, and this is what lets you know that you are moving forwards in a particular direction relative to the world around you. Optic flow data is usually represented using vector fields such that each vector represents the apparent direction and speed of a point in the observer's field of view.

The human visual systems are remarkably good at estimating heading, or the direction of self-motion, from optic flow [7,8]. This ability is what allows people to know what direction they are moving in when they drive down the road or walk down a hallway on a daily basis. Without this visual optic flow data, however, people really quickly become completely disoriented and can no longer tell what direction they are moving in. For example, in the classic experiment when people are asked to walk in a straight line while blind-folded, they are often surprised that they are unable to continue to move straight forwards after only a few steps [17].

Current research centered around creating automated systems, such as self-driving cars, robots, etc, involves creating algorithms that allow machines to use video input data as (optic flow) to determine the direction of their own motion relative to the world around them. It is desirable to build autonomous systems that estimate self-motion from optic flow because it only requires a relatively inexpensive camera(s) and would allow robust navigation without GPS and other large, expensive sensor arrays, such as LiDAR. The human brain demonstrates that an optic flow-only based system can be both fast and accurate [7,8], but in order for these systems to be reliable, we need algorithms that are able to estimate heading from optic flow as accurately as the human brain can. Current algorithms can accurately determine heading based on optic flow video data that is consistent with the observer's motion. However, in the real world, we often come across scenarios where we must be able to accurately determine heading using optic flow that is more complex. One such scenario is when things such as other moving objects enter the observer's field of view and create new motion vectors that are inconsistent with the direction of the observer's motion. For example, if an observer is driving down the street and a person walks across the crosswalk in front of them, the walker adds new motion vectors to the observer's field of view that are inconsistent with the optic flow generated by the observer's own motion. Luckily for us, the human visual system is robustly designed so that we mostly ignore this inconsistent flow data and still know what direction we're moving in when things like other moving objects enter our field of view [9-12]. Machines, on the other hand, have a much harder time with this and large moving objects and other inconsistent flow data tends to throw-off their heading estimation.

Recent work has demonstrated that convolutional neural networks (CNNs) and recurrent neural networks (RNNs) can successfully be applied to problems related to autonomous navigation [1-5] using visual information/sensors. Since optic flow has been shown to be an effective stimulus to estimate self-motion using vision, our approach focuses on heading estimation from optic flow. Our goal was to compare the robustness of heading estimation between CNN and RNN neural network architectures. We used optic flow that varied parametrically in flow density as well as type and amount of inconsistent optic flow to better understand how specific manipulations impacts performance across the networks. We compare the networks' performance with that obtained by human observers under similar circumstances.

Convolutional neural networks CNNs are a class of feedforward deep neural networks that are characterized by architectures with convolutional layers that allow the network to consider the spatial relationships between input pixels in images and videos. Since CNN's have been shown to be ideal networks for most image and video analysis tasks, we decided that a simple CNN would be the ideal architecture to use for our task of estimating heading from optic flow.

Recurrent neural networks (RNNs) are a class of neural networks that consider an ordered sequence of inputs and the temporal relationship between them, rather than considering only one input individually. RNN's are commonly used in natural language processing because when it comes to tasks such as predicting the next word in a phrase, it is important to take into consideration the ordered sequence of words that came before. We hypothesized that a recurrent convolutional neural network may be more robust for our heading estimation task because it can take into consideration the overall pattern of optic flow over time, and thus hopefully ignore inconsistent flow that doesn't fit with this pattern.

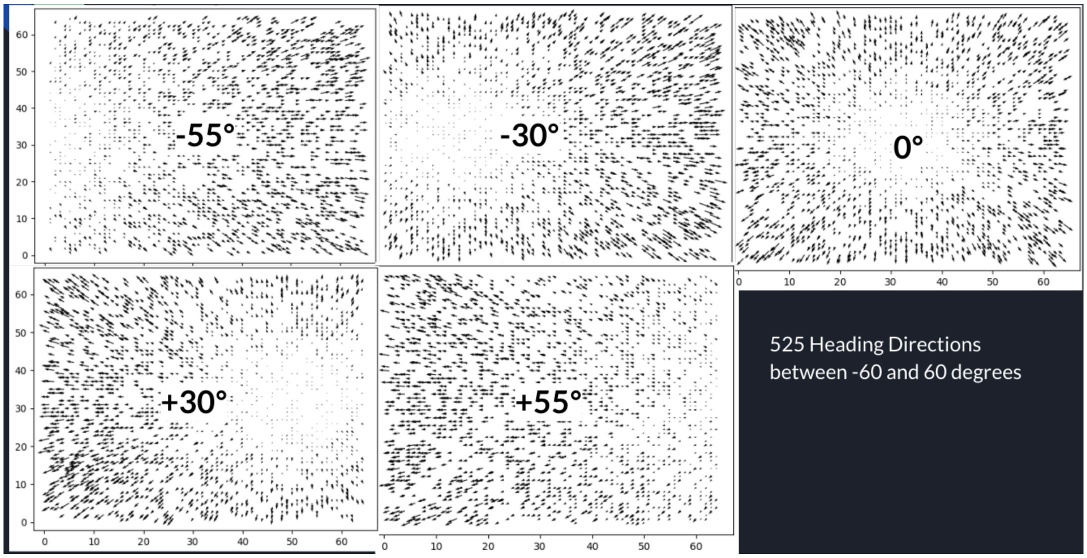
We trained and optimized both a simple convolutional neural network (CNN), and a similar recurrent version (RNN) for the same task of estimating heading from optic flow. The drawback to using an RNN rather than a simple CNN is that RNNs are more complex and take more significantly more time to train. Our goal was to test each of these networks with various examples of inconsistent opic flow data to determine if neural networks can be used to robustly estimate heading from optic flow, and to determine whether the RNN provides any significant advantage to justify the added complexity.

Methods:

Constant Heading Data:

We generated a data set of simulated optic flow with MATLAB and used it to train our models. The data set consists of motion in 525 unique heading directions between -60 and 60 degrees. Zero degrees is equivalent to simulated motion such that the observer is moving straight forwards, while -60 is equivalent to motion in a direction sixty degrees to the observer's left, and +60 is equivalent to motion in a direction sixty degrees to the observer's right. For each of these 525 headings, the data sample consists of 45 temporal frames, each with 2000 simulated motion vectors on a 64x64 grid (see Figure 1 below). We will refer to this data as our "Constant Heading Data Set" since the simulated motion of the observer is in a constant direction for all 45 frames.

Figure 1:



We randomly divided up the constant heading data set into three parts, with 80% of the data being used for training of our models, 10% of the data being used as a validation set during training, and the remaining 10% being set aside and only used to test the performance of trained models.

Training Neural Networks:

We trained our models using the 80% of the constant heading data that was set aside for training, as discussed above. We designed, trained, and tested each model using TensorFlow Keras.

CNN:

Our first model is a CNN regression model that takes in only one static 64x64 frame at a time (with 2000 simulated motion vectors) and predicts the current heading direction based on these vectors. After testing several variations of the model (using separate validation and testing sets) and using grid search to optimize hyperparameters, we determined that we got optimal performance with the following CNN design:

Layers 1 and 2: Convolutional Layers with 128 filters, 7x7 kernel size, and Relu activation
Layers 3: 2x2 Max Pooling Layer

Layers 4 and 5: Convolutional Layers with 128 filters, 5x5 kernel size, and Relu activation
Layers 6: 2x2 Max Pooling Layer

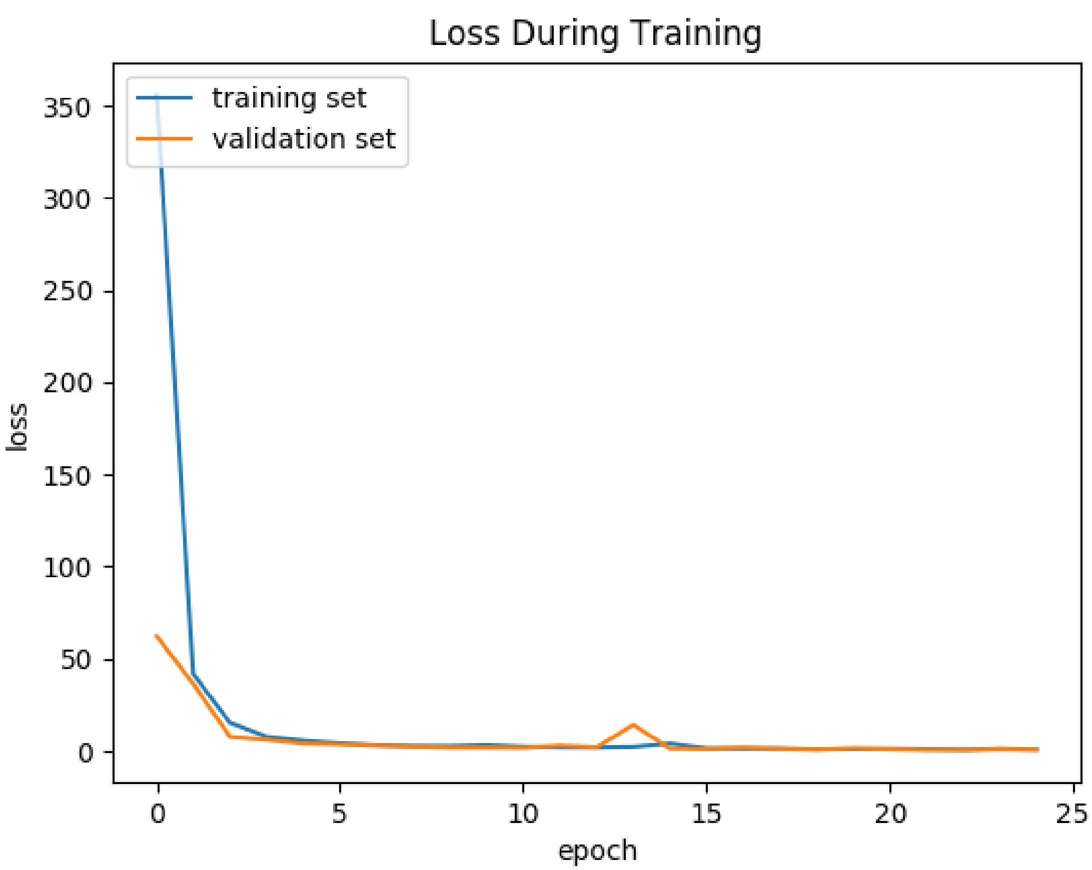
Layers 7 and 8: Convolutional Layers with 512 filters, 3x3 kernel size, and Relu activation
Layers 9: 2x2 Max Pooling Layer

Layers 10 and 11: Convolutional Layers with 1024 filters, 3x3 kernel size, and Relu activation
Layers 12: 2x2 Max Pooling Layer

Layer 13: Fully Connected/ Dense Layer with 128 Nodes and Relu activation
Layer 14: Final Fully Connected/ Dense Layer with one Node and Linear activation

The above simple 14-layer CNN was trained using Adam optimization, a learning rate of 0.0001, and batch sizes of 500. This network was trained for 25 epochs using the constant heading training data to produce our optimal CNN model. The network took approximately 9.23 hours to train using Colby College's Natural Science Computer Cluster of 32 CPUs. Figure 2 shows the loss calculated for both the training set and the validation set after each epoch during the training of our CNN.

Figure 2: Training of Cnn3



RNN:

Next, we trained our RNN. The RNN is a similar regression model used to predict numeric heading direction based on optic flow input. Rather than single static frames, however, the RNN uses ordered sequences of 9 frames of optic flow in a particular heading direction to predict heading. We again tested several variations of the model used simple grid searches to optimize hyperparameters, such that we determined that we got optimal performance with the following RNN design:

Layers 1 and 2: Convolutional Layers with 128 filters, 7x7 kernel size, and Relu activation
Layers 3: 2x2 Max Pooling Layer

Layers 4 and 5: Convolutional Layers with 128 filters, 5x5 kernel size, and Relu activation
Layers 6: 2x2 Max Pooling Layer

Layers 7 and 8: Convolutional Layers with 512 filters, 3x3 kernel size, and Relu activation
Layers 9: 2x2 Max Pooling Layer

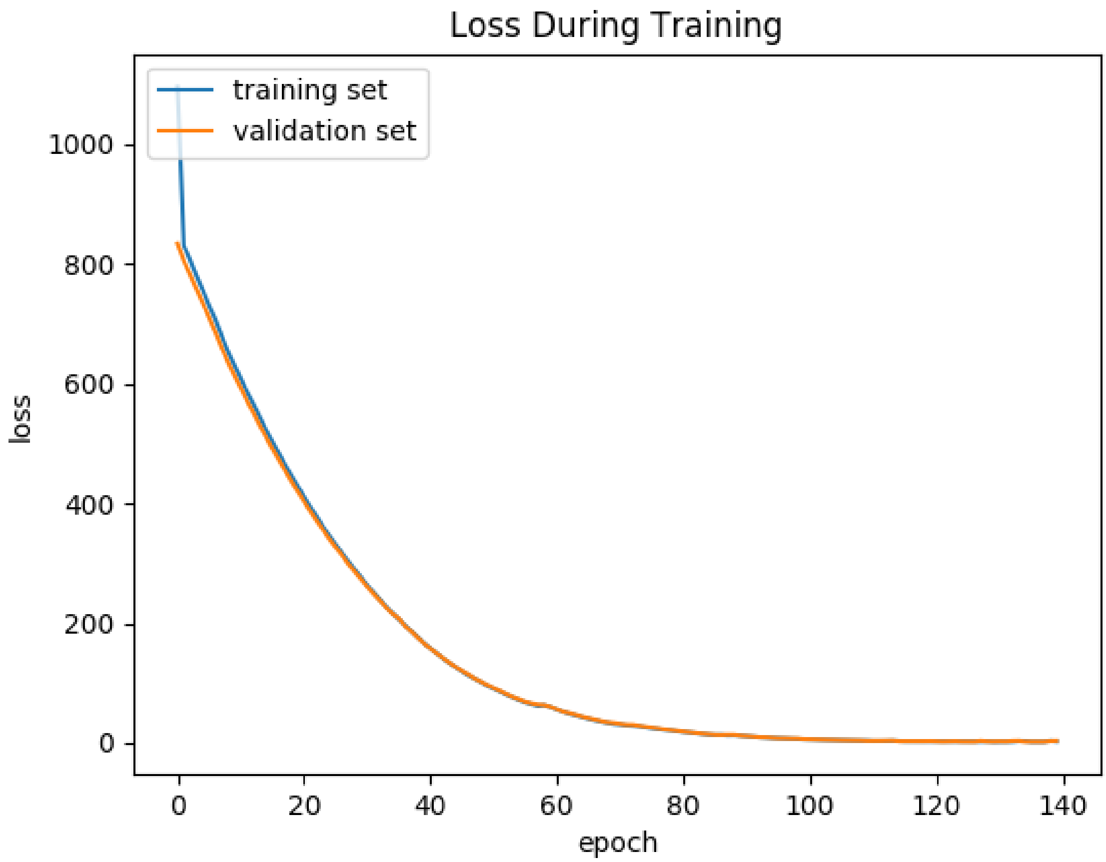
Layers 10 and 11: Convolutional Layers with 1024 filters, 3x3 kernel size, and Relu activation
Layers 12: 2x2 Max Pooling Layer

Layer 16: Long Short-Term Memory Layer with 128 nodes, tanh activation, and sigmoid recurrent activation (see TensorFlow Keras API documentation)

Layer 14: Fully Connected/ Dense Layer with 64 Nodes and Relu activation
Layer 15: Final Fully Connected/ Dense Layer with one Node and Linear activation

The above simple 15-layer RNN was trained using Adam optimization, a learning rate of 0.0001, and batch sizes of 200. This network was trained for 140 epochs using the constant heading training data to produce our optimal RNN model. The network took approximately 45.57 hours to train (considerably longer than the 9.23 hours taken to train the CNN). Figure 3 shows the loss calculated for both the training set and the validation set after each epoch during the training of our RNN.

Figure 2: Training of Rnn1



The Long Short-Term Memory (LSTM) layer is the key layer that allows the above network to consider the temporal relationship between ordered sequences of inputs and thus classifies it as an RNN. Notice that we use equivalent convolutional stacks for both the RNN and the CNN so that we can directly compare their performance to determine how adding the recurrent LSTM layer affects performance.

Results:

In order to evaluate each network's performance, we initially used the 10% of the constant heading optic flow data that we set aside for testing. We compared the heading predicted by each network to the true heading used to generate each data sample. Again, the RNN uses ordered sequences of 9 frames of optic flow data to predict headings, while the CNN must use singular frames. The results of this testing are shown in Figure 3.

Figure 3: Testing with Constant Heading Data

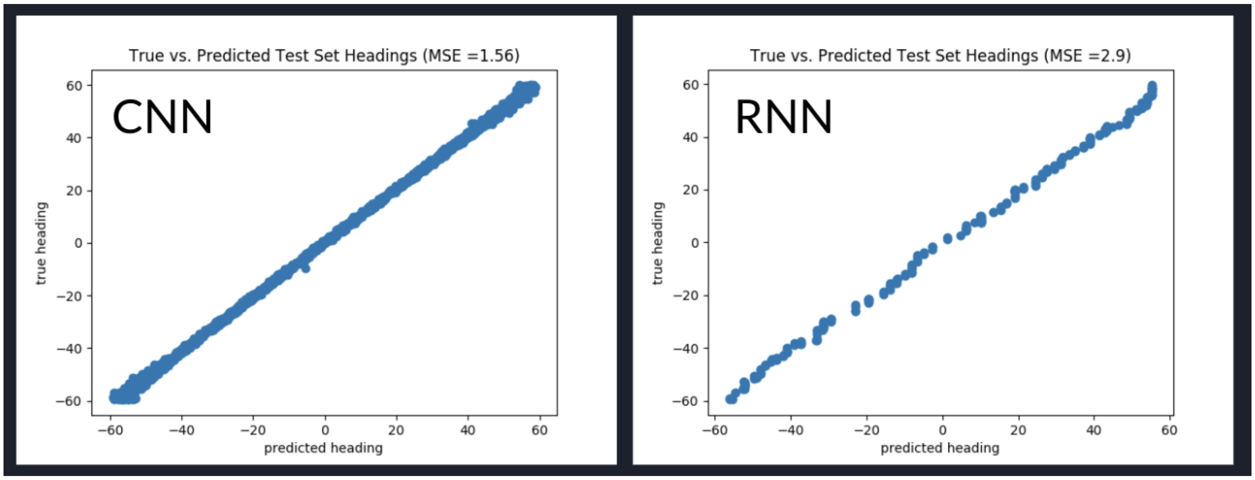


Figure 3: The above scatterplots show each testing data sample as a singular point, with the heading direction predicted by the neural network on the x-axis and the true heading used to generate the optic flow sample on the y-axis.

Notice in Figure 3 that we see minimal deviation between the true heading directions and those predicted by the networks. This indicates that both networks are able to estimate heading from optic flow very accurately in the case when we have simulated opic flow in a constant heading direction with no adversarial/inconsistent flow. Since the human brain can typically estimate heading direction from optic flow accurately to within 1-3 degrees [7,8], and the mean squared error in the heading estimation of both networks is less than 3 degrees, we can conclude that neural networks can be successfully trained to estimate heading from optic flow with human-like accuracy.

Next, we began to test the robustness of each of our models using various examples of optic flow data that is less ideal for heading estimation than the constant heading data set.

Case 1: Sparse Optic Flow Data

Our first test was to see if our networks could still accurately predict heading direction in cases when there is less optic flow data available to them. The human brain can still estimate heading really accurately when there is very little optic flow available. For example, Warren et al. showed that human heading estimation performance remained high with optic flow displays of 63–20 motion vectors [7]. Thus, we want our models to exhibit similar robustness in this case. Recall that the networks were initially trained and tested using our 'constant heading' optic flow data set with 2000 individual motion vectors in each frame. In order to test the robustness of our networks to having less optic flow data to work with, we incrementally decreased the number of motion vectors available to the networks and tested to see how well they could estimate heading. The results of this testing are shown in Figure 4.

Figure 4:

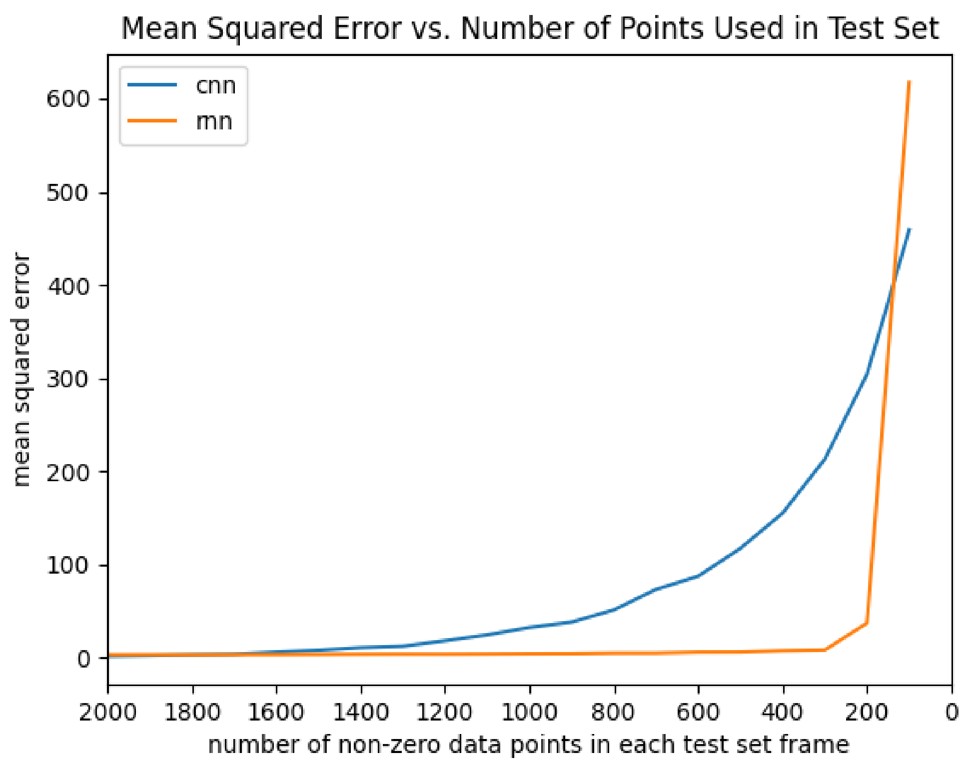


Figure 4: The plot in Figure 4 shows the number of motion vectors available to the network on the x-axis, and the mean-squared error of the network's estimated heading direction on the y-axis. The CNN mean-squared errors are shown by the blue curve and the RNN mean-squared errors are shown by the orange curve.

In Figure 4, notice that the accuracy of the CNN starts to degrade as soon as we decrease the number of motion vectors in each frame to 1000, which is still relatively dense optic flow. This indicates that our CNN model is not nearly as robust as the human visual system when it comes to sparse optic flow. The RNN, on the other hand, is much more robust to sparse optic flow as its accuracy does not begin to degrade until we decrease the number of motion vectors in each frame to 200. This indicates that the RNN is much more robust in scenarios when we have less optic flow data available, thus making a case to justify using an RNN rather than a CNN for heading estimation.

Case 2: 'Drifting Heading'

The next case that we wanted to explore is the case when the observer's heading drifts or changes over time. For the data we initially used to train and test our networks, motion was simulated in a constant heading direction for all 45 frames. So for example, when simulating motion with heading equal to positive 30 degrees, the simulated motion is constant straight-line motion in this direction. However, in reality, people don't usually move in a perfectly straight line when they are walking or driving around and so their heading direction almost always changes, or 'drifts', over time.

To assess the robustness of our models to heading changes, we generated a new data set that simulates optic flow when the observer's heading direction is incrementally changing or 'drifting' from one frame to the next. For example, the heading direction might start at 5 degrees and continuously drift to 10 degrees by the end of the video. We will refer to this as the 'drifting heading' data set. We used this data strictly as a test set to determine the robustness of each of our networks in this scenario. The results of this testing are shown in Figure 5. Note that the CNN and RNN models tested are the same models that we trained and optimized using the constant heading data set as described above.

Figure 5:

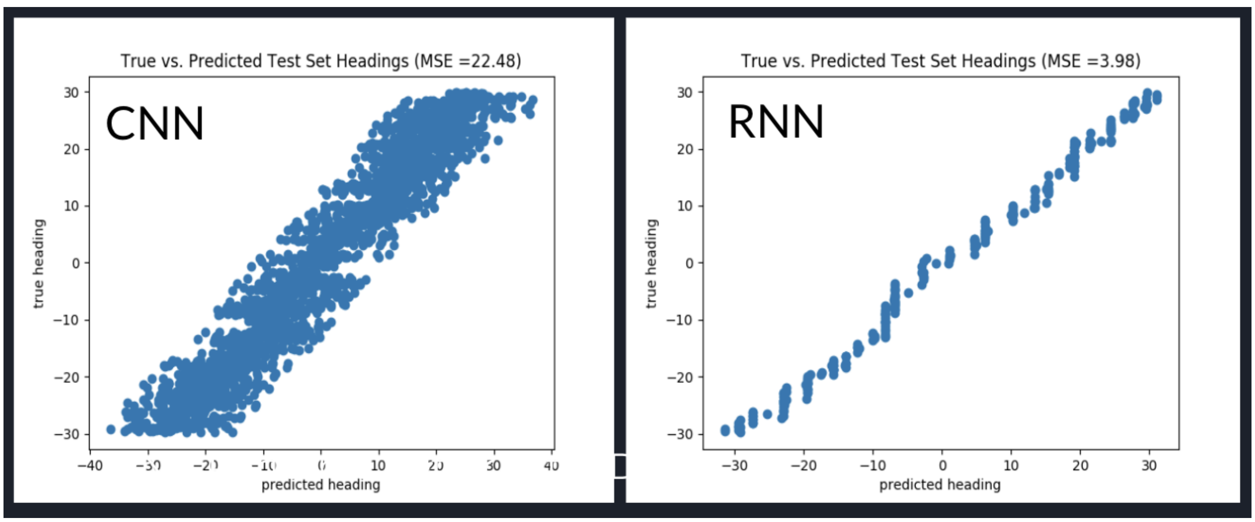


Figure 5: The above scatterplots show each testing data sample from the drifting heading data set as a singular point, with the heading direction predicted by the neural network on the x-axis and the true heading used to generate the optic flow sample on the y-axis.

Notice in Figure 5 that in the RNN scatter plot, we see minimal deviation between the true heading directions and those predicted by the network, while in the CNN scatterplot, we see larger deviations between the true and predicted heading directions. Furthermore, the mean squared error (MSE) in the heading estimation of the RNN is only 3.98 degrees while the MSE for the CNN is 22.48 degrees. This indicates that the RNN is more robust and is able to estimate heading from the 'drifting' optic flow much more accurately than the CNN. This result thus strengthens a justification for using an RNN rather than a CNN for heading estimation. A likely explanation for this is that the RNN is able to take into account the ordered sequence of 9 frames of data so that the fact that the heading is 'drifting' or changing over time, in this case, doesn't impact the overall heading estimation as much.

Case 3: Moving Object

Recall that for the data we used to train our networks, optic flow was simulated assuming that the field of view was stationary so that all simulated motion vectors are a direct result of the motion of the observer. In reality, however, other moving objects often enter the observer's field of view and generate motion vectors that are inconsistent with the motion of the observer. We generated another similar data set that simulates optic flow in 525 constant heading directions, with 2000 additional motion vectors added to simulate a large 20x20 object moving through the observer's field of view. We'll refer to this as the 'large moving object' data set. We used this data to test each of our networks to determine their robustness in this scenario. The results of this testing are shown in Figure 6.

Figure 6:

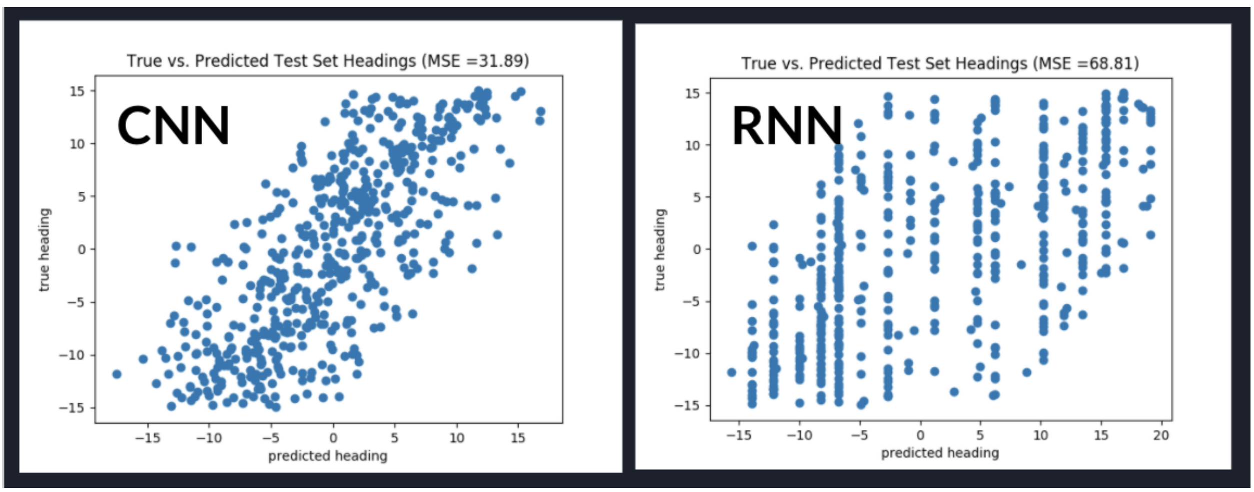


Figure 6: The above scatterplots show each testing data sample from the large moving object data set as a singular point, with the heading direction predicted by the neural network on the x-axis and the true heading used to generate the optic flow sample on the y-axis.

Notice in both scatter plots in Figure 6, we see very large deviations between the true and predicted heading directions. Furthermore, the MSEs in the heading estimation of both networks are above 30 degrees. This indicates that both networks are inaccurate when it comes to estimating heading when from optic flow with a large moving object. Further, the RNN provides no noticeable advantage here.

Moving Object Path Angle:

We also investigated the relationship between the path angle of the simulated object and the ability of the network to estimate heading accurately. Here, the path angle of the object is the angle between the direction of the object's motion and the motion of the observer (see Figure 7).

The large moving object data set has examples of moving objects with various path angles between -25 and 25 degrees. We found that the larger the absolute value of this path angle was for a particular data sample, the larger the error in the heading estimation was for both networks (see Figure 8).

Figure 7:

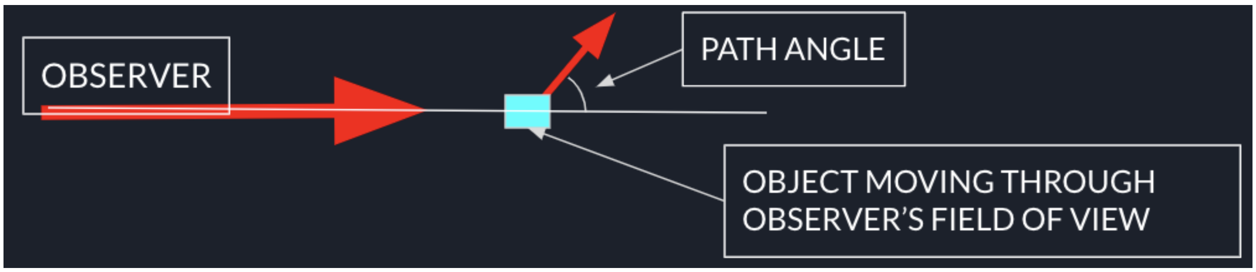


Figure 8:

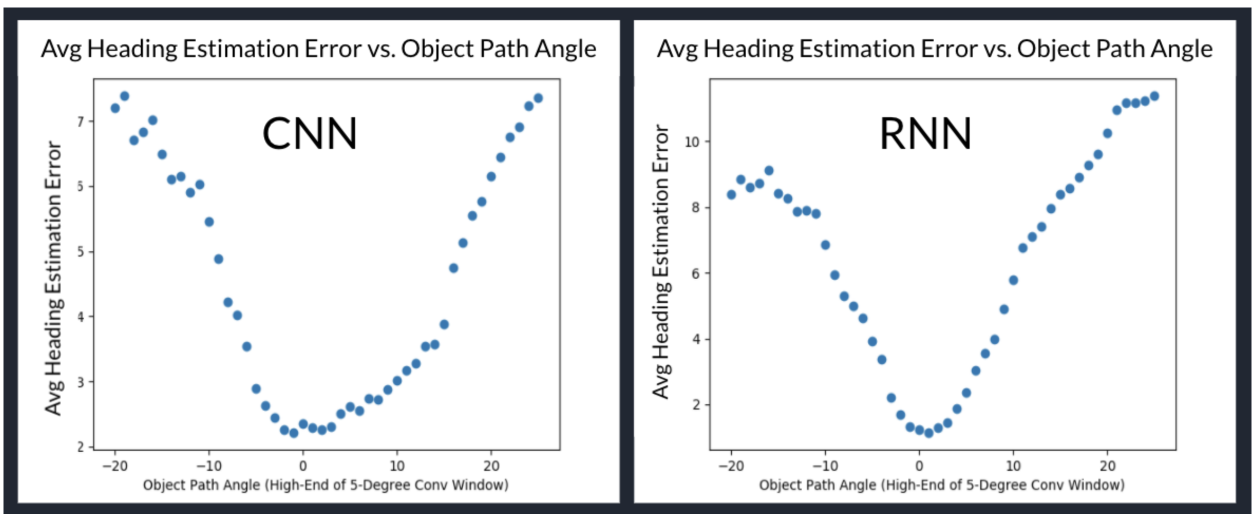


Figure 8: The two scatter plots here show the path angle of the simulated moving object in a given data sample on the x-axis, and the average error in the heading estimation of the network on data samples with that path angle on the y-axis. The "error" in heading estimation is defined as the absolute value of the difference between the true heading direction used to generate the sample and the heading direction estimated by the network (error = |true heading - predicted heading|). Each data point plotted represents the average error over a 5-degree convolutional window of path angles.

In Figure 8, we see a clear u-shape for each network indicating that as the absolute value of the object's path angle increases, the error increases correspondingly. In other words, the larger the object's relative path angle, the more it disrupts the network's ability to correctly estimate heading direction.

Smaller Moving Object:

The ‘large moving object’ data set was generated using a very large simulated object with dimensions of 20x20. This object thus takes up a large percentage of each 64x64 frame in the data. We, therefore, decided to test to see if our models might perform better with a smaller moving object. To test this, Professor Layton generated another data set that simulates optic flow in 525 constant heading directions, with 2000 additional motion vectors added to each frame to simulate a 7x7 ‘medium-sized’ object moving through the observer’s field of view. We’ll refer to this as the ‘medium moving object’ data set. We used this data to test each of our networks to determine their robustness in this scenario. The results of this testing are shown in Figure 9.

Figure 9:

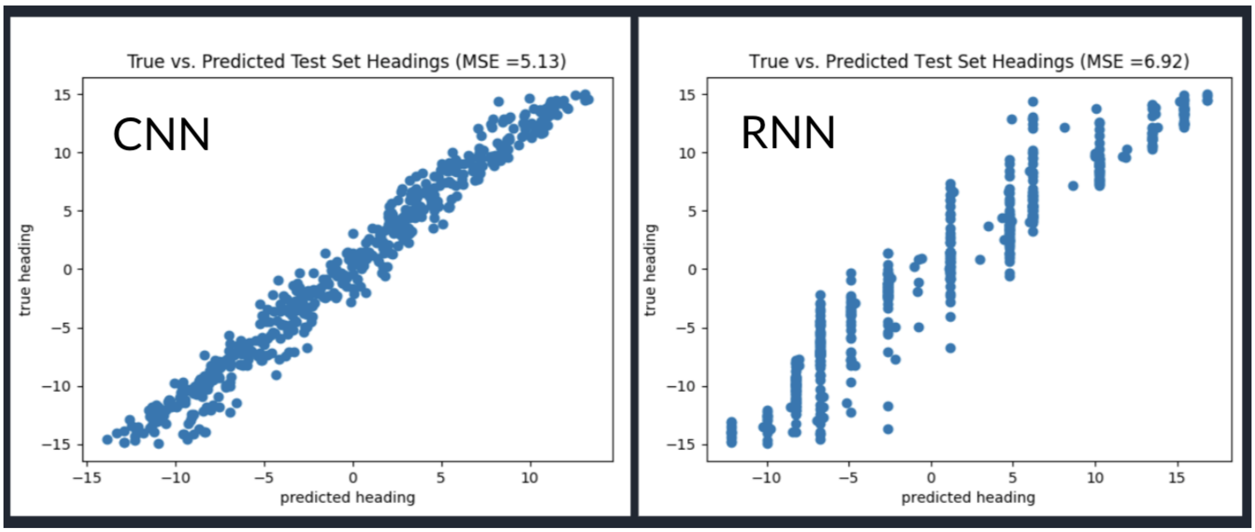


Figure 9: The above scatterplots show each testing data sample from the medium moving object data set as a singular point, with the heading direction predicted by the neural network on the x-axis and the true heading used to generate the optic flow sample on the y-axis.

Notice in both scatter plots in Figure 9, we see much smaller deviations between the true and predicted heading directions than we saw in Figure 6. This indicates that the models are able to estimate heading from optic flow much more accurately when the simulated object is made smaller. Although there is a bit more deviation between the true and predicted heading directions when compared to our initial testing with the constant heading data, the MSEs of the heading estimations of both models are within 5-7 degrees for this data set, indicating that the models are fairly robust to optic flow with this smaller moving object. Further, notice that the RNN does not provide an advantage over the CNN for accurately estimating heading in this case.

Although these results show that our models are fairly robust to smaller moving objects, in order for them to be reliable in real-world scenarios, we would like them to be robust to larger moving objects as well. Thus, we decided to take another approach to attempt to create a model that is more robust to estimating heading from optic flow with large moving objects.

Handling Large Moving Objects (Another Approach):

The small moving object dataset results demonstrated that CNNs and RNNs can estimate heading with only modest increases in error compared to constant flow. This prompted us to explore the degree to which the networks could learn heading well in the presence of large moving objects that occupy most of the visual field. To address this question, we introduce a subset of the large moving object samples into the training set.

Training with Moving Objects:

We trained a new CNN, this time using a combination of our original constant heading data set along with data samples from our 'large moving object data set'. We used a data set with a total of 500 training samples, with 250 taken from the constant heading data set and 250 taken from the large moving object data set. A random 10% of this new data set was used as a validation set, with the remaining 90% used to train our new CNN model. In training this new model, we used the exact same network architecture, training time, optimization, learning rate, batch size, etc., such that the only difference between this new model and our original CNN model is the data used to train each of them. Figure 10 shows the loss calculated for both the training set and the validation set after each epoch during the training of this model.

Figure 10: Training New CNN with Examples of Large Moving Objects

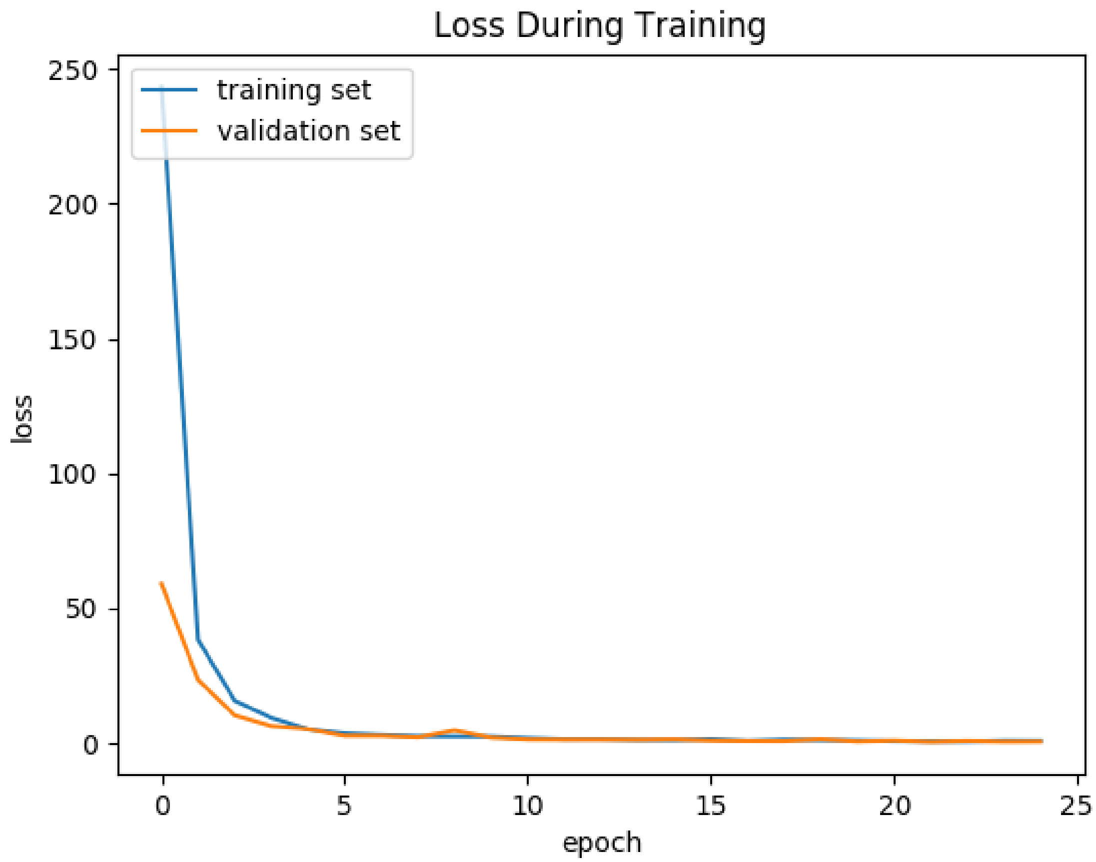


Figure 10: The plot in Figure 10 shows the loss calculated for both the training set and the validation set after each epoch during the training of our CNN models. The training and validation sets in this case are composed of data such that half of the examples are from the constant heading data set and half are from the large moving object data set.

After training, we tested the model using data from the large moving object data set that was never shown to the model during training. The results of this testing are shown in Figure 11.

Figure 11:

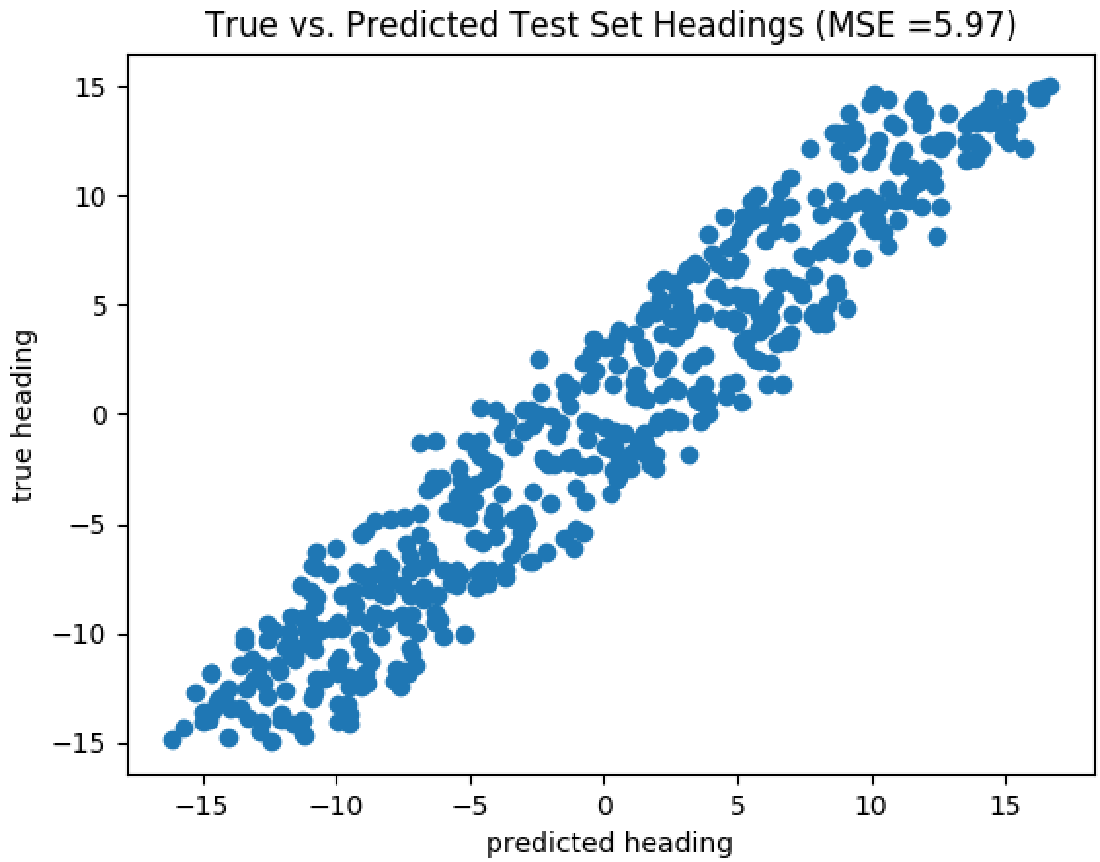


Figure 11: The above scatterplots show each testing data sample from the portion of the large moving object data set aside for testing as a singular point, with the heading direction predicted by the new CNN model on the x-axis and the true heading used to generate the optic flow sample on the y-axis.

Notice in Figure 11 that we see much smaller deviations between the true and predicted heading directions than we saw in Figure 6, and a much smaller MSE of only 6 degrees rather than over 30 degrees. This result confirms our hypothesis that training a CNN with data including examples of large moving objects drastically increases the robustness of the model such that it can estimate heading from optic flow with large moving objects fairly accurately. Thus, this training strategy presents a promising solution to handling optic flow with large moving objects.

Recall that in order to train this new CNN, we used a training data set with 250 of the samples taken from the constant heading data set and the other 250 taken from the large moving object data set. We wondered if it was actually necessary to include such a large number of examples of moving objects in our training data such that half of the training data contained a large moving object. We are currently working on training new models with a larger number of samples from our original constant heading optic flow data, and fewer samples from our large moving object data, to try to determine how many examples of moving objects are actually necessary to train models that are robust to scenarios when they encounter moving objects.

Case 4: Rotation

The next case that we wanted to explore is the case when the eye of the observer rotates as the observer moves through the world. For all of the simulated optic flow data we have generated and tested our models with thus far, we have assumed pure translation such that the radial flow pattern is a direct result of the motion of the observer. But in reality, the eye of the observer may also be rotating in different directions relative to the observer him/herself as he/she moves. For example, as you walk down the street you might turn your head or glance around in different directions as you move so that the focus of expansions no longer aligns with your true heading direction; human heading estimation is poor under these conditions [13-16]. Thus, we wanted to test to see whether the heading estimations of our models would be similarly affected in this case.

To assess the robustness of our models to rotation, we generated a new data set that is similar to the original constant heading data set in that we simulate optic flow in 567 heading directions between -60 and 60 degrees, such that the heading direction remains constant for all 45 frames. However, for this new data set, we simulate the alteration of the observed motion vectors resulting from simulated rotation of the eye of the observer from one frame to the next. We will refer to this as the 'rotation' data set. We used this data to test each of our networks to determine their robustness in this scenario. The results of this testing are shown in Figure 12.

Figure 12:

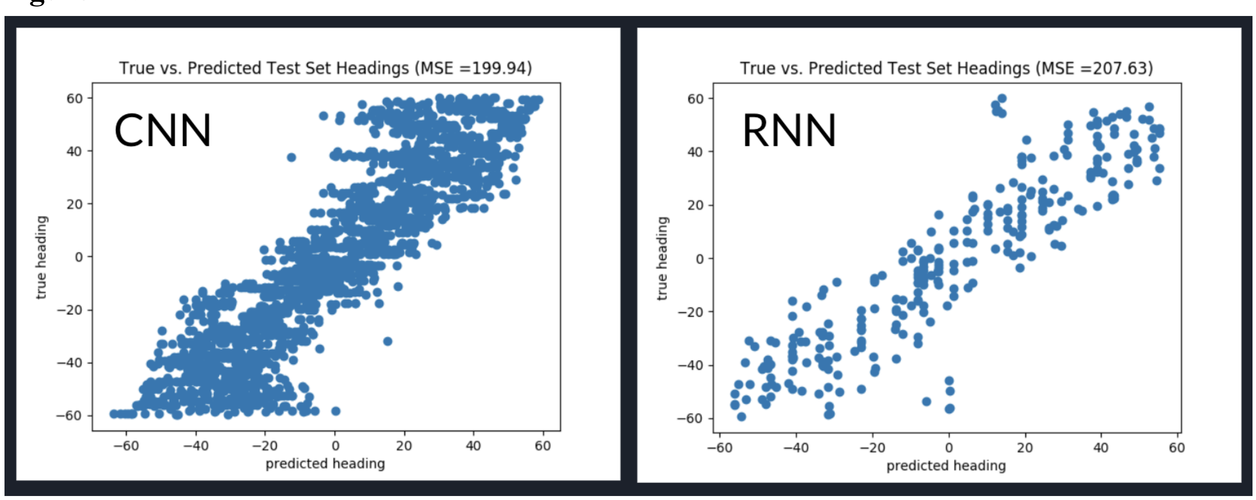


Figure 12: The above scatterplots show each testing data sample from the rotation data set as a singular point, with the heading direction predicted by the neural network on the x-axis and the true heading used to generate the optic flow sample on the y-axis.

Notice in both scatter plots in Figure 12, we see large deviations between the true and predicted heading directions. This indicates that both networks are inaccurate when it comes to estimating heading when from optic flow with simulated rotation.

Impact of Rotation Magnitude on Heading Estimation:

When testing our models with the rotation data, we found that the magnitude of the rotation vector used to simulate the rotation of the eye of the observer has a large impact on the ability of the models to estimate heading accurately. In our rotation data set, rotation was simulated in various different directions in each sample such that the magnitude of the rotation vectors used were randomized between 0 and 0.4 degrees. We found that the larger the magnitude of this rotation vector was for a particular data sample, the larger the error in the heading estimation was for both models (see Figure 13).

Figure 13:

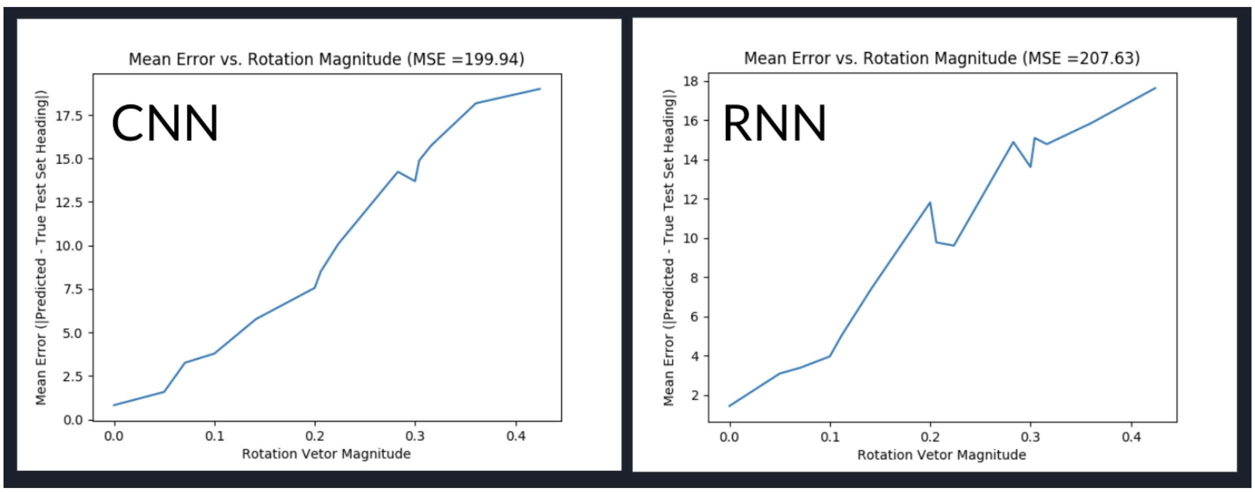


Figure 13: The two plots here show the rotation vector used plotted on the x-axis, and the error in the heading estimation of the model on the y-axis (averaged over all test samples in the rotation data set which used rotation vectors with the same magnitude). All numeric values are in degrees.

For both networks, we see a clear trend such that the average error in heading estimation rapidly increases as we increase the magnitude of the rotation vector. In other words, the more that the eye of the observer rotates from one frame to the next, the more the simulated rotation deters each model's ability to correctly estimate heading direction. This result is analogous to our previous result that increasing the relative path angle of a moving object correspondingly degrades the ability of our models to estimate heading direction accurately.

Handling Rotation (Another Approach):

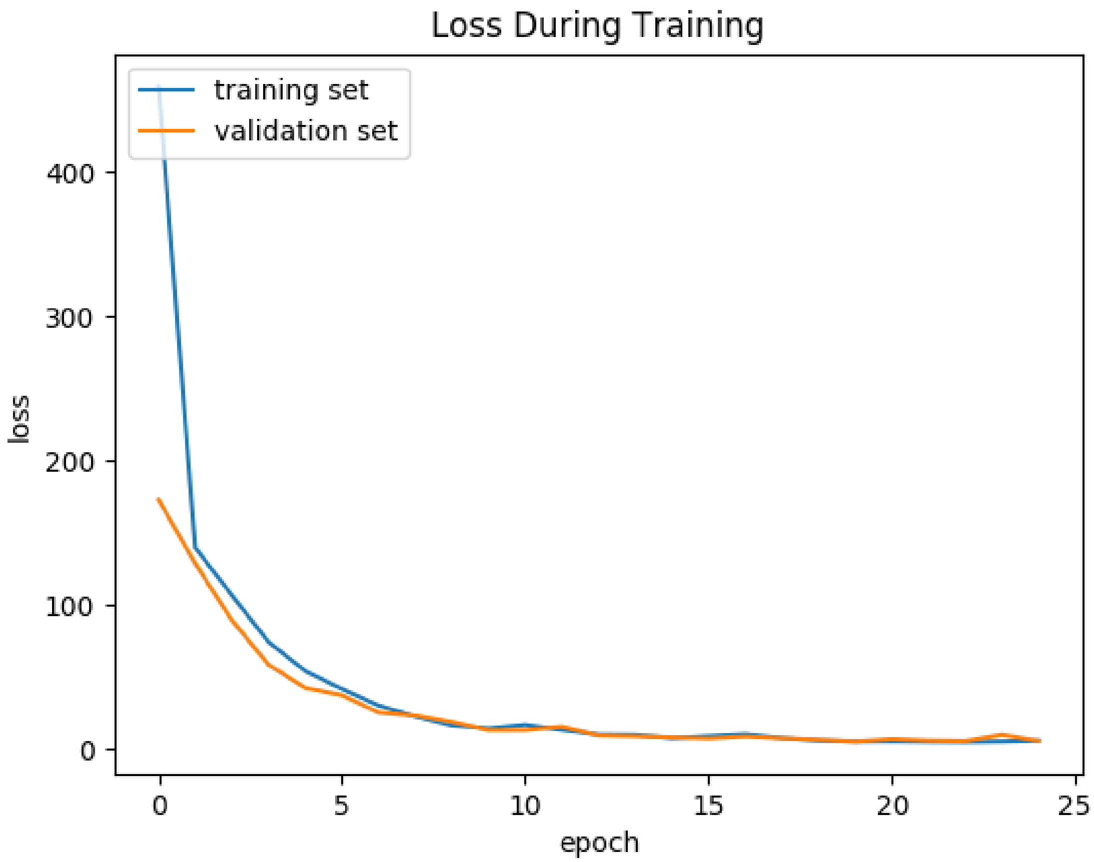
Since neither of our models was able to accurately estimate heading from optic flow with rotation, we decided to take a similar approach to the one we took to handle the case of large moving objects, and train a new CNN using training examples with rotation. We hypothesized that showing the CNN examples of optic flow that contains rotation during training could allow the network to become more robust, just as training our ‘objects CNN’ with examples of moving objects allowed it to more accurately estimate heading when presented with other optic flow containing moving objects.

Since both models performed poorly in estimating heading from rotating optic flow, it follows that using an RNN does not provide any advantage over the CNN in this particular case to justify the added complexity and training time. We thus decided only to train a CNN in this case.

Rotation CNN:

We trained a new CNN model, which we’ll call ‘rotation CNN’, using a training data set composed of 50% of the samples from the rotation data set and 50% of the samples from the original constant heading data set. The remaining 50% of the rotation data samples were kept explicitly separate so that they could be used to test the accuracy of the resultant model. The network we train to create our rotation CNN model was designed with the same exact layers as our original CNN model and was similarly trained for 25 epochs using Adam optimization, a learning rate of 0.0001, and batch sizes of 500. Figure 14 shows the loss calculated for both the training set and the validation set after each epoch during the training of the model.

Figure 14: Rotation CNN Training



Next, we tested the rotation CNN model using the remaining data from the rotation data set that we set aside for testing. The results of this testing are shown in Figure 15.

Figure 15:

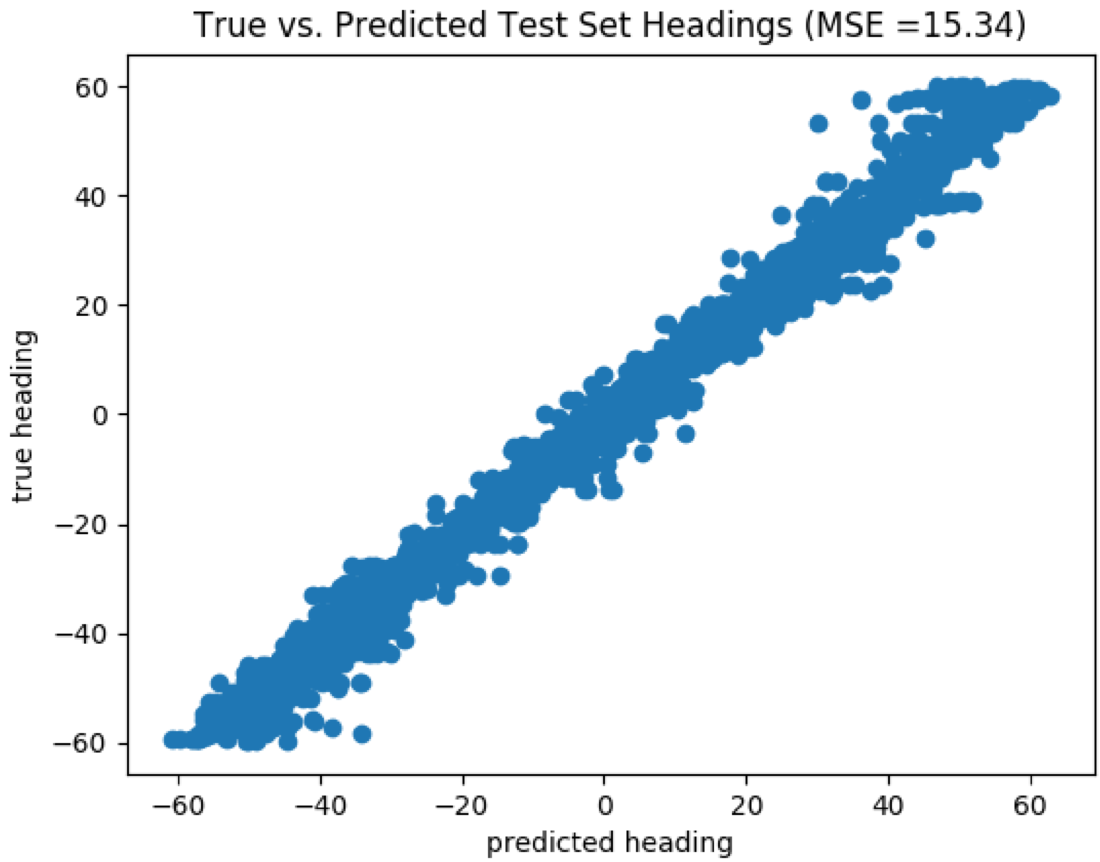


Figure 15: The above scatterplot shows each testing data sample from the portion of the rotation data set aside for testing as a singular point, with the heading direction predicted by the 'rotation CNN' model on the x-axis and the true heading used to generate the optic flow sample on the y-axis.

Notice in Figure 15 that the rotation CNN model that we trained with examples of rotation is able to estimate heading from rotating optic flow much more accurately than the original CNN model. However, there is still some significant deviation between the true and predicted heading directions and the MSE in the heading estimation is still 15.34 degrees. This indicates that the robustness of a CNN to rotation can indeed be improved by adding examples of rotation to the training data set, but that our rotation CNN model fails to predict heading as accurately when presented with optic flow with simulated rotation. It is possible that adding more examples of rotation to the training data set could further improve the model's performance in this case, and this is something that we plan to continue to explore in the future.

Conclusion:

We found that both a simple convolutional neural network and a recurrent neural network can be used to effectively estimate heading from optic flow with human-like accuracy. Additionally, we found that the recurrent neural network is the ideal choice for robustly estimating heading from optic flow in some more realistic scenarios - in particular in the cases when we have sparse optic flow data and when we have heading that is drifting or changing rather than remaining constant over time. However, we found that both models perform poorly and that using an RNN provides no additional robustness in the additional realistic case when we have optic flow with large moving objects, and in the case when we have rotating optic flow. However, we found that showing our models examples of optic flow with large moving objects and rotating flow improves its robustness in estimating heading in each of these cases.

Overall, our analysis shows that neural networks can be trained to estimate heading direction from optic flow with similar accuracy and robustness to humans. Thus, neural networks present a promising method to estimate heading from video optic flow data so that we can create autonomous systems that can move through the world without the need for expensive sensors or other navigation systems. In the future, we plan on training and testing networks with more realistic optic flow video data from various environments in order to create models that could allow for robust navigation in the real world.

Citations:

[1] Zhu, A. Z., Yuan, L., Chaney, K., & Daniilidis, K. (2019). Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 989-997).

[2] Costante, G., Mancini, M., Valigi, P., & Ciarfuglia, T. A. (2015). Exploring representation learning with cnns for frame-to-frame ego-motion estimation. *IEEE robotics and automation letters*, *1*(1), 18-25.

[3] Q. Zhao, F. Li and X. Liu, "Real-Time Visual Odometry Based on Optical Flow and Depth Learning," *2018 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, 2018, pp. 239-242, doi: 10.1109/ICMTMA.2018.00064.

[4] Pandey, T., Pena, D., Byrne, J., & Moloney, D. (2021). Leveraging deep learning for visual odometry using optical flow. *Sensors*, *21*(4), 1313.

[5] Y. Xu, Y. Wang and L. Guo, "Unsupervised Ego-Motion and Dense Depth Estimation with Monocular Video," *2018 IEEE 18th International Conference on Communication Technology (ICCT)*, 2018, pp. 1306-1310, doi: 10.1109/ICCT.2018.8600039.

[6] P. Muller and A. Savakis, "Flowdometry: An Optical Flow and Deep Learning Based Approach to Visual Odometry," *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017, pp. 624-631, doi: 10.1109/WACV.2017.75.

[7] Warren, W. H., Morris, M. W., & Kalish, M. (1988). Perception of translational heading from optical flow. *Journal of Experimental Psychology: Human Perception and Performance, 14*(4), 646–660. https://doi.org/10.1037/0096-1523.14.4.646

[8] A.V. van den Berg, Robustness of perception of heading from optic flow, Vision Research, 1992

[9] Royden, C. S., & Hildreth, E. C. (1996). Human heading judgments in the presence of moving objects. *Perception & psychophysics*, *58*(6), 836-856.

[10] Warren Jr, W. H., & Saunders, J. A. (1995). Perceiving heading in the presence of moving objects. *Perception*, *24*(3), 315-331.

[11] Layton, O. W., & Fajen, B. R. (2016). The temporal dynamics of heading perception in the presence of moving objects. *Journal of Neurophysiology*, *115*(1), 286-300.

[12] Layton, O. W., & Fajen, B. R. (2016). Sources of bias in the perception of heading in the presence of moving objects: Object-based and border-based discrepancies. *Journal of Vision*, *16*(1), 9-9.

[13] Longuet-Higgins Hugh Christopher and Prazdny K.. 1980 The interpretation of a moving retinal image *Proc. R. Soc. Lond. B.*208385–397

[14] Warren, W. H., & Hannon, D. J. (1990). Eye movements and optical flow. *JOSA A*, *7*(1), 160-169.

[15] Royden, C. S., Banks, M. S., & Crowell, J. A. (1992). The perception of heading during eye movements. *Nature*, *360*(6404), 583-585.

[16] Li, L., & Warren Jr, W. H. (2000). Perception of heading during rotation: Sufficiency of dense motion parallax and reference objects. *Vision research*, *40*(28), 3873-3894.

[17] McManus, C. *Right Hand, Left Hand: the Origins of Asymmetry in Brains, Bodies, Atoms and Cultures*. Phoenix, 2004.