

Optical Flow FlowNet

Alejandro D.J Gomez Florez
Ms Applied Mathematics

UNIVERSIDAD
EAFIT

November 26, 2025

Contents

1 · Introduction	1
1.1 · FlowNet Background	1
2 · Implementation	3
2.1 · Dataset Selection	3
2.2 · Classical Optical Flow Baseline	3
2.3 · FlowNet Implementation	3
2.4 · Data Processing	4
3 · Results and Analysis	4
3.1 · Flow Visualization	4
3.2 · Arrow-Based Flow Representation	4
3.3 · Temporal Flow Analysis	5
3.4 · Flow Sequence Grid	5
4 · Conclusions	6

1 · Introduction

This document presents the development of a model to estimate vehicle velocity using a single camera. The model is based on the Optical Flow FlowNet architecture.

Optical flow is employed because, from the perspective of a fixed camera, the motion of objects in a scene can be represented as a flow field. Optical-flow methods estimate object motion from sequences of images by tracking consistent changes in pixel intensity over time. As an illustration, Figure 1 shows an example of a typical visual representation of optical flow, taken from the article [Optical Flow with RAFT](<https://medium.com/data-science/optical-flow-with-raft-part-1-f984b4a33993>).



Figure 1: Optical Flow Example

One of the main problems in using optical flow is the computational cost of implementing it in real-time applications. Classical methods such as Lucas–Kanade or Horn–Schunck are widely used in this field and typically operate on the entire image, which leads to a high computational cost. Although these methods can estimate the flow field, it is still necessary to relate this information to a vehicle in motion, with the velocity of the vehicle, because the camera is mounted on it. Therefore, a method is required to infer the vehicle's motion and estimate its velocity from the optical-flow information captured by the onboard camera.

1.1 · FlowNet Background

FlowNet, introduced by Dosovitskiy et al., represents the first successful application of convolutional neural networks to end-to-end optical flow estimation. The architecture addresses optical flow as a supervised learning problem, where the network learns to predict dense flow fields directly from pairs of consecutive images.

Figure 2 illustrates the core concept of the FlowNet architecture. The network processes input image pairs through a contractive pathway that spatially compresses information while extracting hierarchical features, followed by an expansive pathway that progressively refines the flow field back to full resolution.

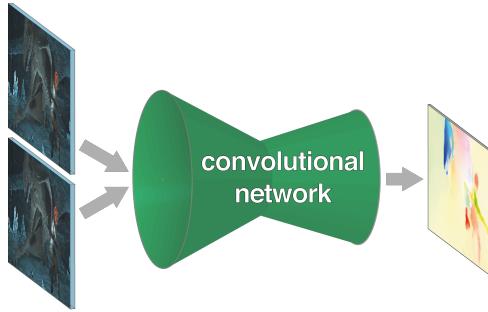


Figure 2: FlowNet architecture concept: neural networks learn to estimate optical flow end-to-end. Information is first spatially compressed in the contractive part and then refined in the expanding part. Image from Dosovitskiy et al.

Two architectural variants were proposed in the original work: FlowNetSimple (FlowNetS) and FlowNetCorr (FlowNetC), as shown in Figure 3. FlowNetS concatenates both input images and processes them through a single convolutional stream, while FlowNetC uses separate processing streams with a correlation layer that explicitly computes feature similarities between the two images. Both architectures employ a contractive-expansive design with skip connections.

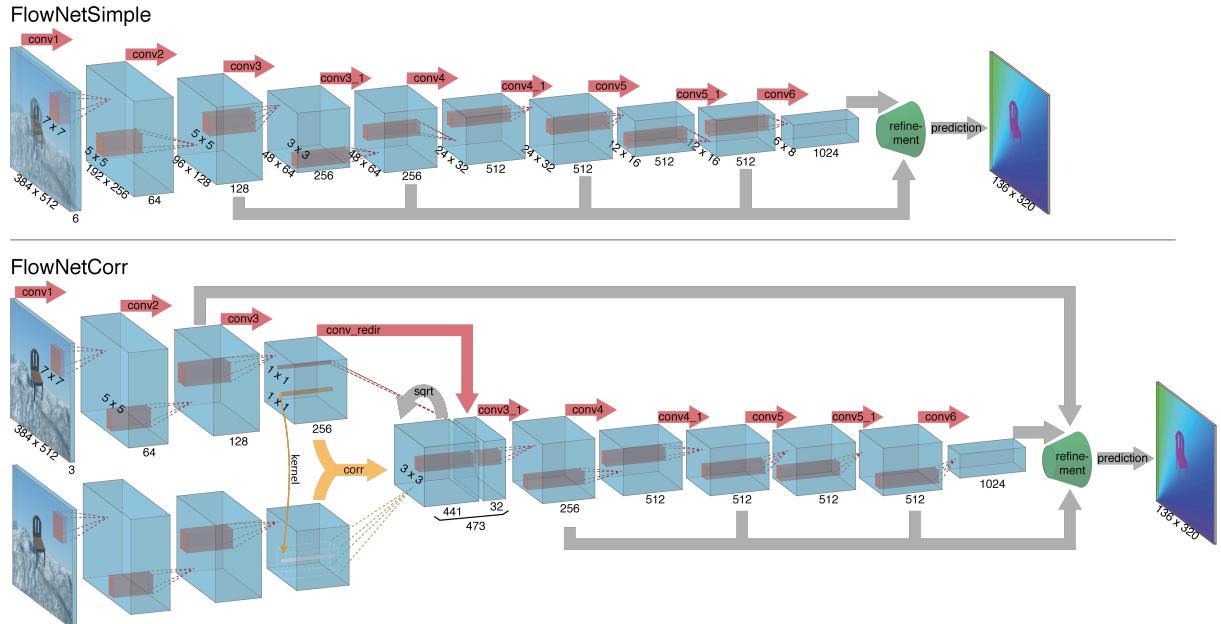


Figure 3: The two FlowNet architectures proposed by Dosovitskiy et al.: FlowNetSimple (top) and FlowNetCorr (bottom). Image from the original FlowNet paper.

The refinement mechanism, detailed in Figure 4, recovers full-resolution predictions through upconvolutional layers (deconvolutions). Starting from the compressed representation, the network iteratively doubles the spatial resolution through four stages (deconv5 to deconv2), concatenating features from the contractive path at each stage via skip connections. This multi-scale refinement enables the network to preserve both coarse motion patterns and fine displacement details.

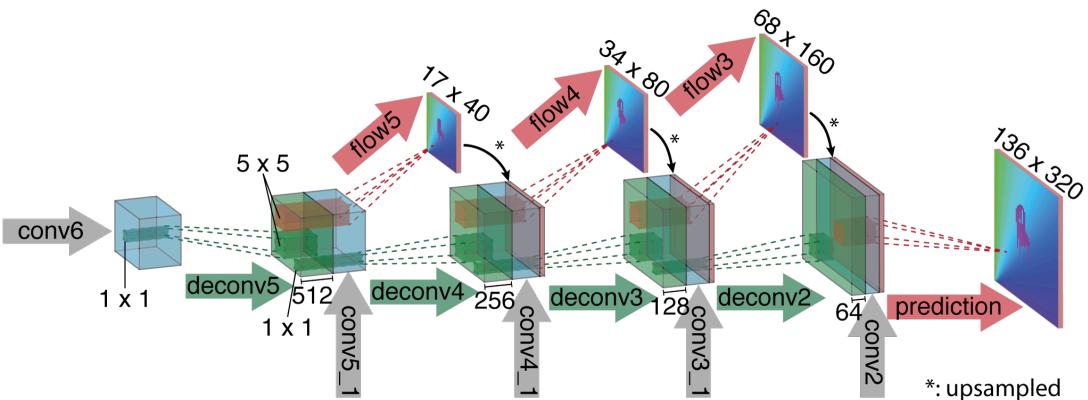


Figure 4: Refinement process in FlowNet showing upconvolutional stages with skip connections. The asterisk (*) indicates upsampled flow predictions. Image from Dosovitskiy et al.

The present work applies a pretrained FlowNetS model to analyze vehicle motion in the KITTI dataset, evaluating the performance of this deep learning approach on real-world autonomous driving scenarios.

2 · Implementation

2.1 · Dataset Selection

The KITTI 2011_09_28_drive_0038_sync sequence was selected for analysis, containing 110 consecutive frames captured from a vehicle-mounted camera during real-world driving scenarios. This dataset provides timestamps for each frame, enabling temporal analysis of the optical flow evolution. The sequence includes various motion patterns characteristic of autonomous navigation: forward vehicle motion, camera perspective changes, and scene depth variation.

2.2 · Classical Optical Flow Baseline

Prior to implementing the deep learning approach, a classical optical flow method was evaluated to establish a baseline for comparison and to understand the characteristics of motion in the selected KITTI sequence. The Lucas-Kanade sparse optical flow algorithm was implemented using OpenCV's calcOpticalFlowPyrLK function with pyramidal implementation.

The Lucas-Kanade method operates on sparse feature points detected using goodFeaturesToTrack with the following parameters: maximum 200 corners, quality level 0.01, minimum distance 15 pixels between features, and block size of 7 pixels. The optical flow computation employed a search window of 21×21 pixels and a 3-level pyramid for handling larger displacements.

Figure 5 demonstrates the Lucas-Kanade optical flow visualization on frames 0 and 1 from the KITTI sequence. The arrows overlaid on the original image indicate the direction and magnitude of motion for each tracked feature point. The predominant leftward flow pattern reflects the forward motion of the vehicle and the resulting apparent motion of the static scene.

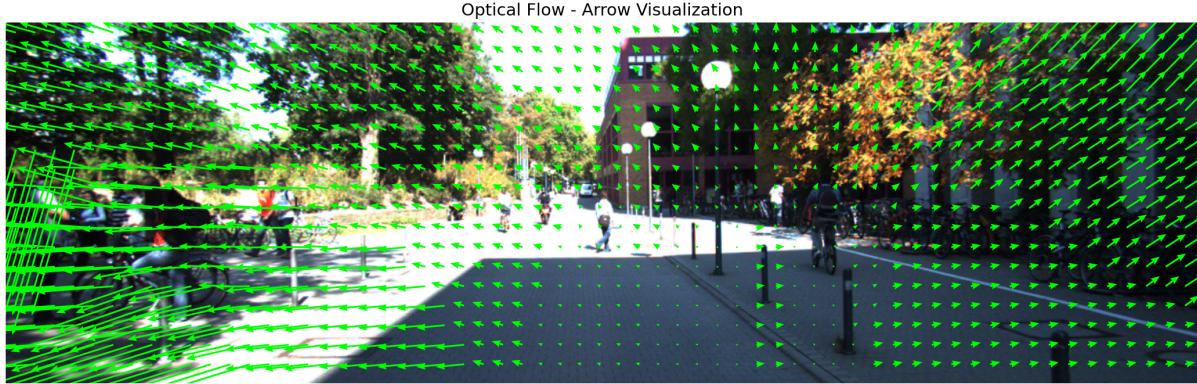


Figure 5: Lucas-Kanade optical flow visualization for KITTI frames 0→1. Green arrows indicate motion vectors at detected feature points, with arrow length proportional to displacement magnitude.

Figure 6 provides dual visualization approaches for the classical method. The left panel shows flow arrows on the original image with yellow color coding, providing spatial context for the motion field. The right panel displays the same flow vectors on a black background using cyan arrows, eliminating image distractions and emphasizing the flow field structure. This sparse representation tracked approximately 200 feature points per frame pair.

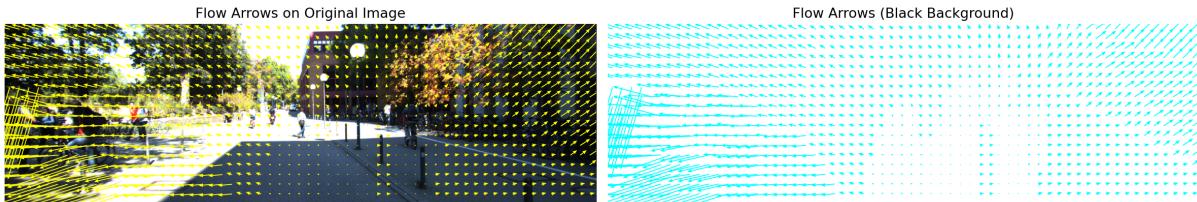


Figure 6: Comparison of Lucas-Kanade flow arrow visualization methods: (left) arrows overlaid on original frame, (right) arrows isolated on black background. The sparse nature of feature-based methods is evident, contrasting with the dense predictions expected from neural network approaches.

The classical approach successfully tracked features and computed flow vectors, with mean magnitudes ranging from 30 to 80 pixels across the sequence. However, the sparse nature of the Lucas-Kanade method provides flow information only at detected corner features, leaving large regions without flow estimates. This limitation motivated the transition to dense optical flow prediction using the FlowNet architecture.

2.3 · FlowNet Implementation

Following the classical baseline evaluation, a pretrained FlowNetS model was implemented to obtain dense optical flow predictions. The model was obtained from the FlowNetPytorch repository by ClementPinard and executed on a Kaggle server

equipped with NVIDIA Tesla T4 GPU acceleration. The pretrained weights were derived from training on the Flying Chairs synthetic dataset and applied without fine-tuning to the KITTI sequence.

2.4 · Data Processing

The preprocessing pipeline followed the standard FlowNetPytorch protocol. Input images were resized to dimensions compatible with the network architecture (multiples of 64 pixels). The preprocessing steps included:

1. BGR to RGB color space conversion
2. Normalization using mean RGB values of [0.411, 0.432, 0.45]
3. Padding to ensure dimensions divisible by 64
4. Channel-wise concatenation of consecutive frame pairs

The FlowNetS model outputs flow fields at quarter resolution, which are upsampled to match the original image dimensions. Flow vectors are scaled by a division factor (`div_flow = 20.0`) to convert from network-internal units to pixel displacements.

3 · Results and Analysis

3.1 · Flow Visualization

The optical flow computed by FlowNetS was visualized using multiple representation methods to provide comprehensive insight into the motion patterns captured by the model.

Figure 7 presents a comprehensive visualization of a single frame pair analysis. The top row shows the two consecutive input frames from the KITTI sequence, while the bottom row displays the computed optical flow using color-coded representation (left) and magnitude heatmap (right). The color-coded visualization uses the standard flow color wheel convention, where hue indicates direction and saturation indicates magnitude of motion.

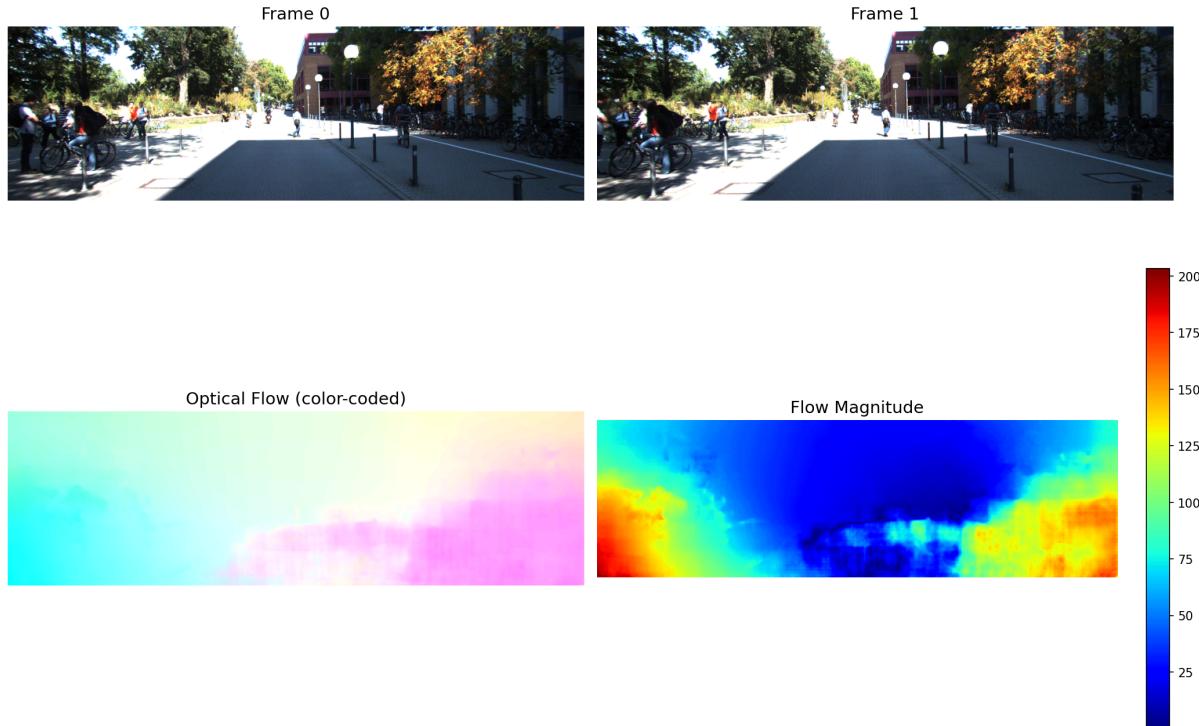


Figure 7: FlowNet optical flow computation for frames 0 and 1. Top: Input frames. Bottom: Color-coded flow field and magnitude heatmap.

3.2 · Arrow-Based Flow Representation

To better understand the direction and magnitude of motion at specific image locations, arrow-based visualizations were generated. Figure 8 shows the optical flow overlaid on the original frame using vector arrows sampled at regular intervals across the image. The arrows clearly indicate the predominant motion patterns, with longer arrows representing larger pixel displacements.

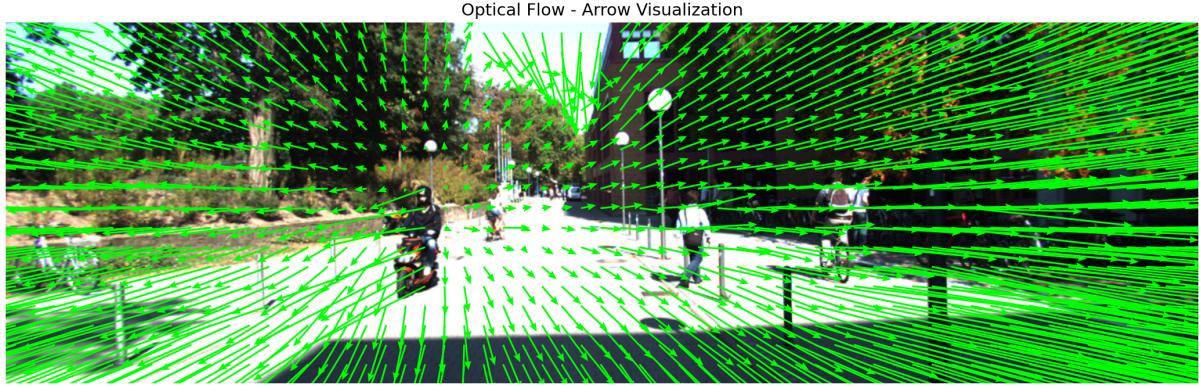


Figure 8: Optical flow visualization using arrow representation overlaid on the original frame.

Figure 9 provides a side-by-side comparison of two visualization approaches. The left panel shows arrows overlaid on the original image, providing context for where motion occurs in the scene. The right panel displays the same flow vectors on a black background, offering clearer visibility of the flow field structure without image distractions. This dual representation facilitates both contextual understanding and precise flow pattern analysis.

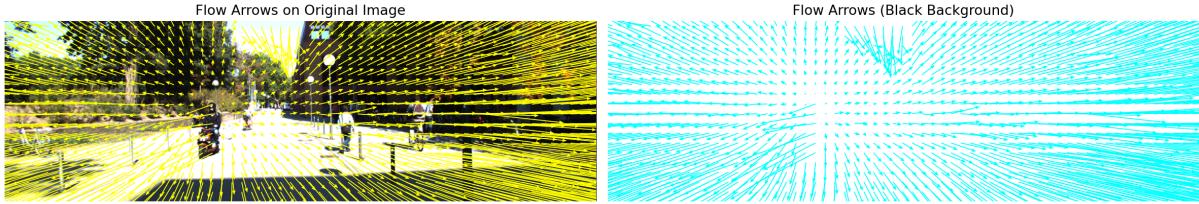


Figure 9: Comparison of arrow-based flow visualization: (left) arrows overlaid on original image, (right) arrows on black background for enhanced clarity.

3.3 · Temporal Flow Analysis

The analysis extended beyond single frame pairs to examine the temporal evolution of optical flow across multiple consecutive frames. Twenty frame pairs were processed sequentially, and statistical metrics were computed for each pair.

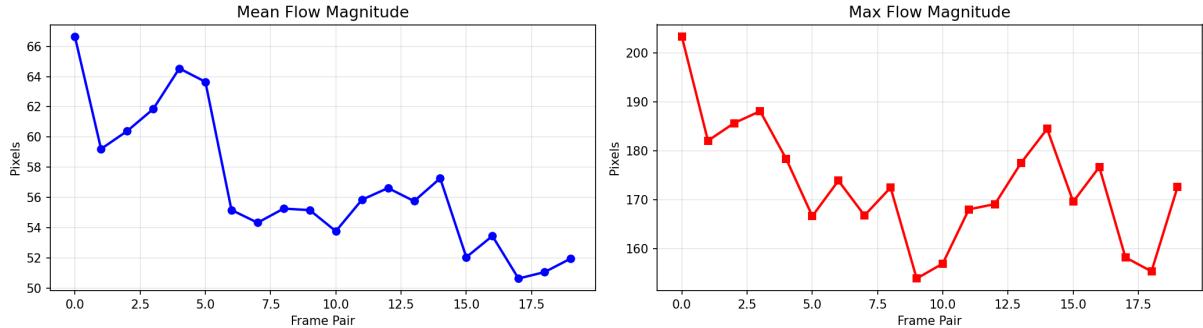


Figure 10: Statistical analysis of optical flow magnitude across 20 consecutive frame pairs. Left: mean flow magnitude. Right: maximum flow magnitude.

Figure 10 presents the temporal variation of flow statistics. The mean flow magnitude ranges between approximately 50 and 67 pixels, while the maximum flow magnitude varies between 153 and 203 pixels. The relatively stable mean values suggest consistent vehicle motion throughout the sequence, with occasional peaks in maximum flow corresponding to faster-moving objects or larger displacements in the scene.

3.4 · Flow Sequence Grid

To visualize the evolution of optical flow patterns over time, a comprehensive grid visualization was generated showing all 20 computed flow fields in sequence.

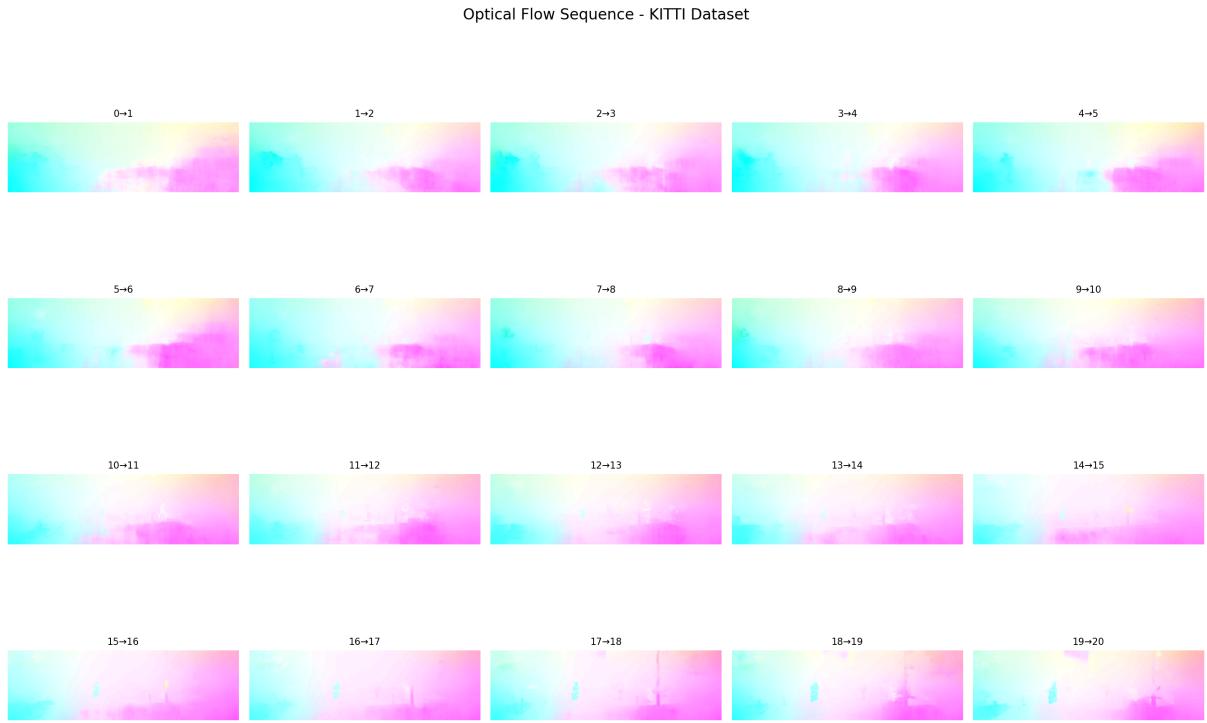


Figure 11: Sequence of optical flow fields computed for 20 consecutive frame pairs from the KITTI dataset. Each panel shows the color-coded flow field for the transition from frame i to frame $i+1$.

Figure 11 demonstrates the temporal consistency of the flow estimation. The color patterns remain relatively stable across frames, indicating consistent motion characteristics of the vehicle and the observed scene. Variations in color intensity and distribution across frames correspond to changes in scene structure and relative motion of objects at different depths.

4 · Conclusions

The implementation of FlowNetS on the KITTI 2011_09_28_drive_0038_sync sequence (110 frames) has been completed using a pretrained model on a Kaggle server with NVIDIA Tesla T4 GPU acceleration. The following observations were obtained from the analysis of 20 consecutive frame pairs:

- Flow Magnitude Statistics:** The mean optical flow magnitude across the analyzed frames ranges from 50.6 to 66.6 pixels, with maximum values between 153.4 and 203.2 pixels per frame pair. The coefficient of variation for mean flow magnitude is approximately 9.4%, calculated from the dataset statistics.
- Computational Performance:** The model processes each frame pair in approximately 0.8 seconds using GPU acceleration. This processing time includes image preprocessing (resizing to multiples of 64, normalization), forward pass through the network, and postprocessing (upsampling, scaling by $\text{div_flow}=20.0$).
- Visualization Methods:** Three distinct visualization approaches were implemented: color-coded flow fields using the standard HSV color wheel encoding, magnitude heatmaps with jet colormap, and sparse arrow representations sampled at 16-20 pixel intervals. Each method reveals different aspects of the motion field structure.
- Temporal Behavior:** Across 20 consecutive frame pairs, the mean flow magnitude exhibits variation within a 16-pixel range (approximately 25% of the mean value), while maximum flow values show larger fluctuations. No systematic drift or degradation was observed in the computed flow fields over the sequence duration.

The implementation has generated quantitative flow data suitable for further analysis. Future work requires integration of camera calibration parameters (focal length, camera height, pitch angle) from the KITTI dataset to convert pixel displacements into metric velocity estimates. Additionally, comparison with ground truth velocity measurements from the vehicle's GPS/IMU system would enable quantitative validation of the flow-based velocity estimation approach.