Master in
Computer Vision
Barcelona

Project
Module 6
Coordination

Video Surveillance for Road Traffic Monitoring

J. Ruiz-Hidalgo / X. Giró

j.ruiz@upc.edu / ramon.morros@upc.edu

Week 1: Report

Master in
Computer Vision
Barcelona

Teams & Repos (please declare any change)

Repo	Students
Team 1	Rachid Boukir, Josep Bravo, Alex Martín, Guillem Martinez, Miquel Romero
Team 2	Álvaro Budria, Alex Carrillo, Sergi Masip, Adrià Molina
Team 3	Alberto Barreiro, Manel Guzmán, Jiaqiang Ye Zhu and Advait Dixit
Team 4	**Missing names!**
Team 5	Razvan-Florin Apatean, Michell Vargas, Kyryl Dubovetskyi, Ayan Banerjee and Iñigo Auzmendi
Team 6	Guillem Capellera, Ana Harris, Johnny Núñez, Anna Oliveras

Scoring Rubric

Task	Description	Max. Score
T1.1	IoU & mAP for ground truth + noise	2
T1.2	mAp over detections	2
T2	IoU over time	2
T3.1	MSEN	1
T3.2	PEPN	1
T3.3	Analysis & Visualizations	1
T4	Optical Flow Plot	1

T1.1 IoU & mAP for (ground truth + noise)

- Generate noisy annotation to the ground truth data by:
  - Adding to the size and position of bounding boxes
  - Add probability to generate/delete bounding boxes
- Compute the IoU and mAP.
- Study the effect of the parameters governing the noise in your results.

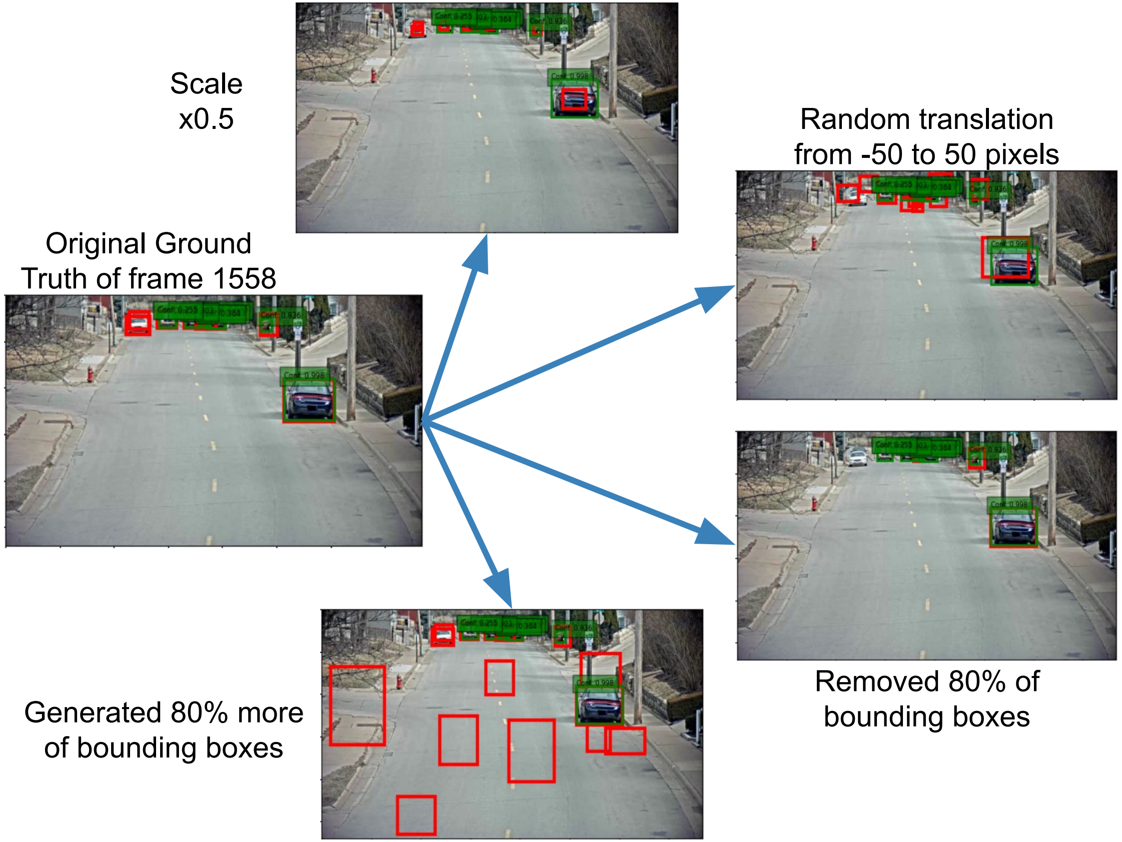
T1.1 IoU & mAP for (Team X) - Copy & paste (max 3. pages)

T1.1 IoU & mAP for (Team 1) - [1/3]

The mean Average Precision metric used in this experiments is based on Pascal VOC. The results presented were extracted sorting the detections by their confidence, and just to compare the possible results that we would have in the case we do not have the confidence score available, we performed a small comparison between confidence sorted and random sorted (average of 10 runs).

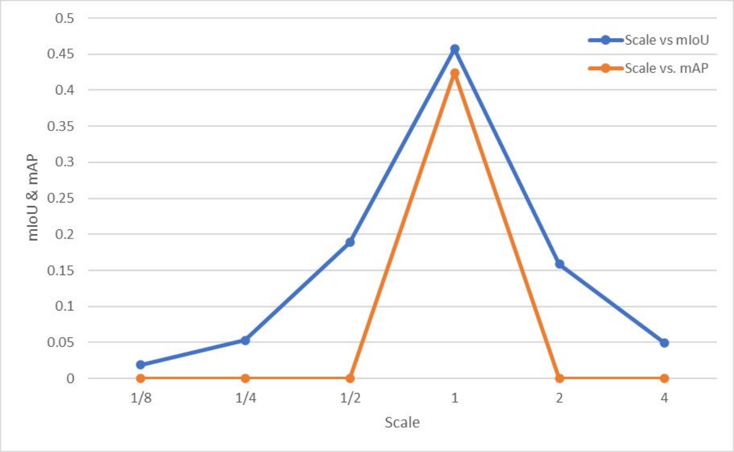
Results obtained with mask RCNN:

mAP sorting	By confidence	Randomly + average with 10 runs
Original Ground truth	0.42	0.28
Scale x0.5	0.00	0.00
Translation 50 px	0.09	0.00
Removing 80% bb	0.36	0.00
Adding 80% bb	0.26	0.20



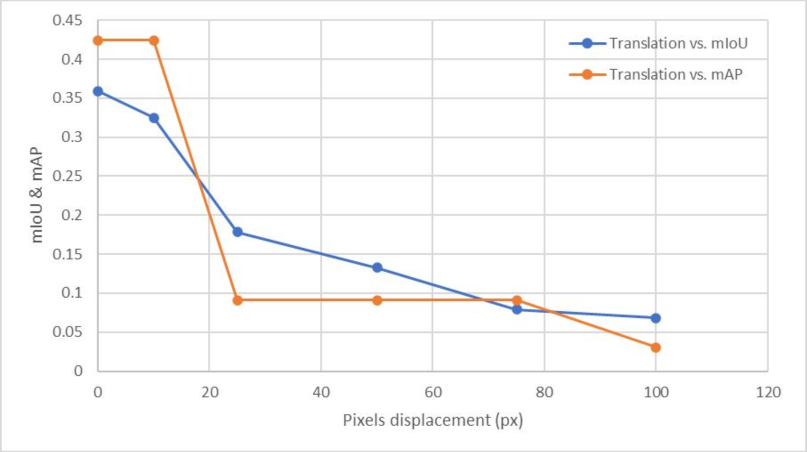
T1.1 IoU & mAP for (Team 1) - [2/3]

Noise added to the **size** of the bounding boxes



We can observe that both increasing and reducing the size of the bounding box, have consequences both in the mIoU and in the mAP.
In the case of reducing the size, is due to the fact that the overlap area is reduced while the are of union remains constant, and in the case of increasing the size we are increasing the area of union while the overlap area remains constant.
The mAP metric is calculated based on the IoU threshold used for evaluating the detections, so in this case, the threshold is not higher than the new IoU, so mAP becomes 0.

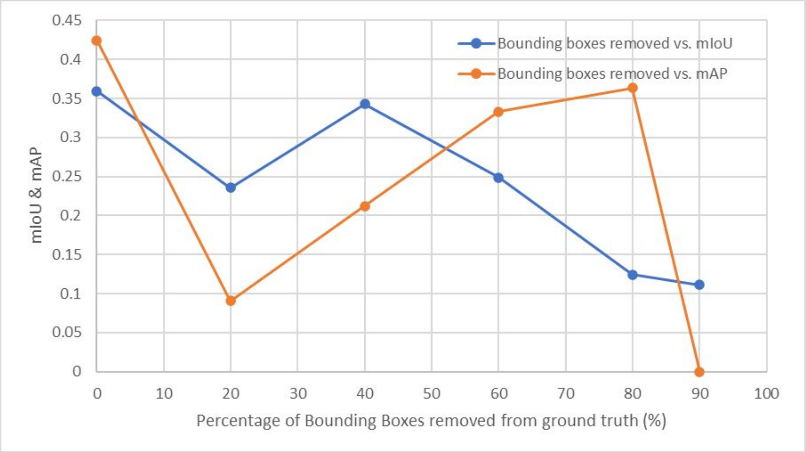
Noise added to the **position** of the bounding boxes



When we change the position of the bounding boxes, we can see that as more pixels the ground truth is moved, lower is the mIoU and mAP, as in this case we are reducing the area of overlap and increasing the area of union.

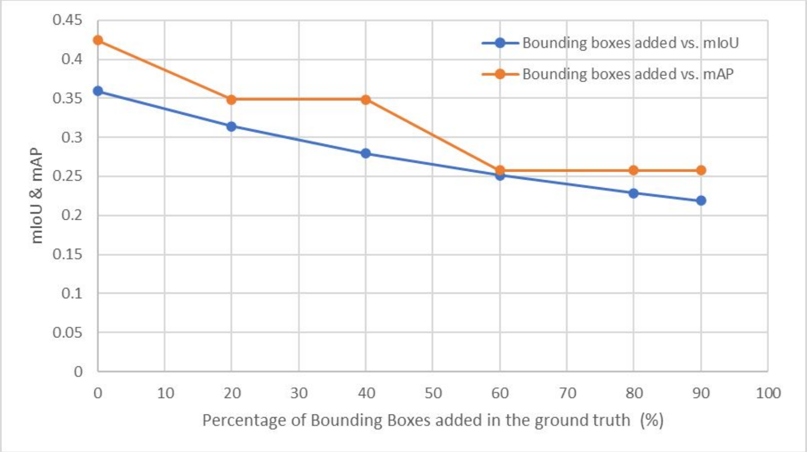
T1.1 IoU & mAP for (Team 1) - [3/3]

Noise added by **removing** bounding boxes



By removing bounding boxes from the ground truth, we can see that there is no clear tendency, as it depends on the bounding boxes removed if they affect the number of false positives, then the mIoU it would increase, and the inverse effect will occur if it affects the true positives with a high IoU. In the case of the mAP, it increases probably because we are removing bounding boxes that they were favouring the number of false negatives, and false positives with high confidence.

Noise added by **generating new** bounding boxes



When we add new bounding boxes in the ground truth, both metrics present a clear tendency to decrease in relation of the percentage of bounding boxes generated. This effect is due to the increment of the quantity of false negatives in the detection.

T1.1 IoU & mAP for (Team 2) - [1/3]

noise distribution	std size	std position	prob delete	prob similar	std similar	min random	max random	similar statistics	mIoU	mAP
Normal	0.1	0.1	0.3	0.1	0.2	0	1	None	.6257	.5426
	0.3	0.1	0.3	0.1	0.2	0	1	None	.5112	.3844
	0.1	0.3	0.3	0.1	0.2	0	1	None	.4104	.0979
	0.1	0.1	0.9	0.1	0.2	0	1	None	.4680	.0909
	0.1	0.1	0.3	0.3	0.2	0	1	None	.5538	.4797

We study the effect of the noise by first setting an arbitrary baseline (green row) of parameters. We experiment with adding normal noise to the size and the position of the ground truth bounding box separately, removing ground truth boxes and adding similar boxes (an extra distorted ground truth box) with a certain probability. In addition, we also add a random number of boxes along the image.

Increasing the noise in the **size** leads to a **drop in both the mIoU and mAP**.

Increasing the noise in the **position lowers the mIoU and the mAP plummets**. This may be explained by the fact that **the mean IoU is below the threshold (AP@0.5)**.

More dropped boxes **lower the mIoU and the mAP plummets**. Now there are **gt boxes without correspondences (FN)**, but some **gt and similar boxes remain**, which keep the mIoU from falling apart.

Adding an **extra similar box** with more noise **slightly lowers both metrics**. Setting a **higher std for the size and position noise of the similar boxes** may affect the **IoU**, while the mAP decreases since we are adding **FPs**.

Notes: The std is in terms of the percentage of the parameter to change (size, position, etc.). Noise in similar boxes is applied to the coordinates directly. Seed=42.

T1.1 IoU & mAP for (Team 2) - [2/3]

noise distribution	std size	std position	prob delete	prob similar	std similar	min random	max random	similar statistics	IoU	mAP
Normal	0.1	0.1	0.3	0.1	0.2	0	1	None	.6257	.5426
	0.1	0.1	0.3	0.1	0.4	0	1	None	.6197	.5425
	0.1	0.1	0.3	0.1	0.05	0	1	None	.6523	.5438
	0.1	0.1	0.3	0.1	0.2	1	1	None	.5926	.5174
	0.1	0.1	0.3	0.1	0.2	0	9	None	.4323	.3974
	0.1	0.1	0.3	0.1	0.2	0	1	mean	.6244	.5421

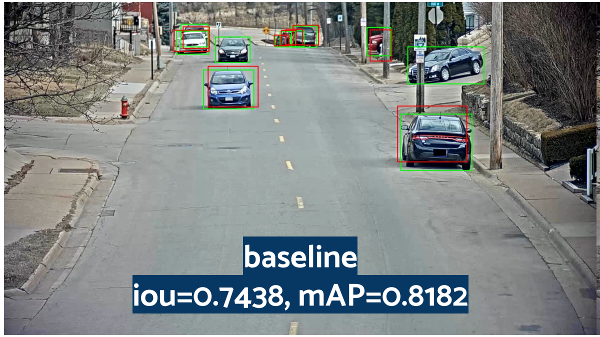
Increasing the **noise in the similar box** barely affects the metrics. However, reducing it slightly improves the IoU. Both **FP and TP proportions** seem to be **unaltered**, but **new and more similar boxes push the IoU up**.

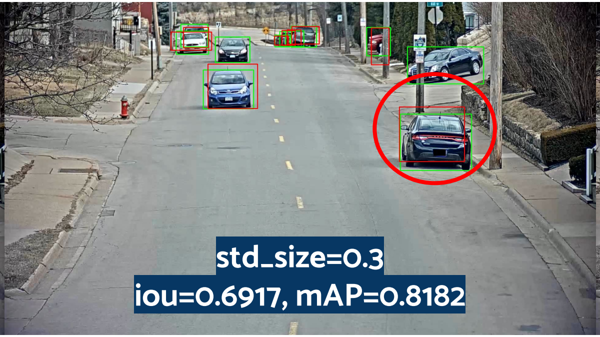
Always adding a **random box slightly decreases both metrics. Increasing the number of random boxes punishes the metrics more**. In both cases we are increasing the number of **FPs**.

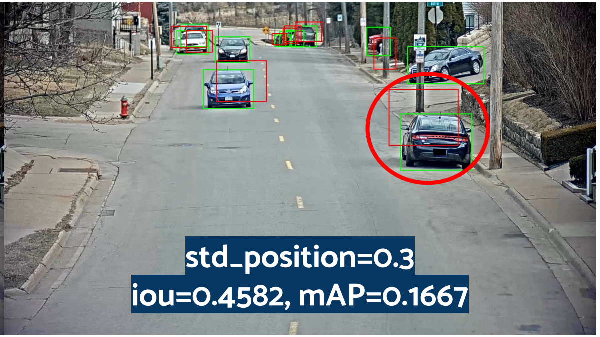
Adding a **size prior** (similar statistics "mean", they are set to have the mean size) to the random boxes (only the position is randomized), **does not have any impact on the metrics**.

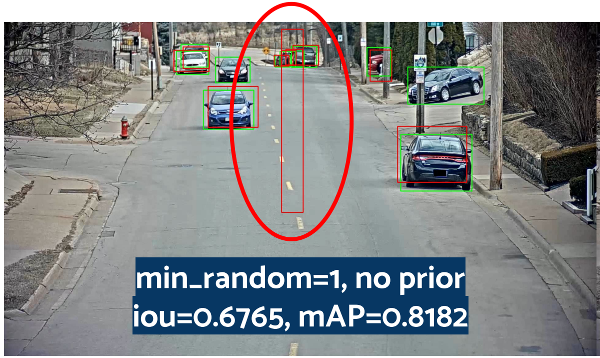
T1.1 IoU & mAP for (Team 2) - [3/3]

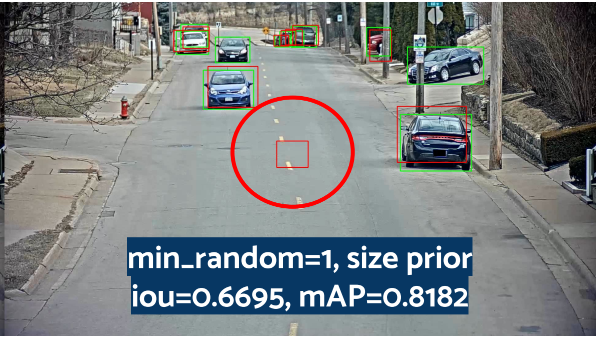
After studying the quantitative results, we visualize the frame #500 with different noise configurations. We compare the baseline against changing the noise of the size and the position. Additionally, we visualize a case with a random box with and without the size prior.











We can see that adding more noise to the position leads to some of the bounding boxes being displaced too much from the ground truth. We also observe that the amount of noise added to the size barely affects it, we would need to increase it even more to notice a drop in the performance. In the last 2 images, we can see that the box with a prior is more natural. Interestingly, one false positive isn't enough to lower the mAP.

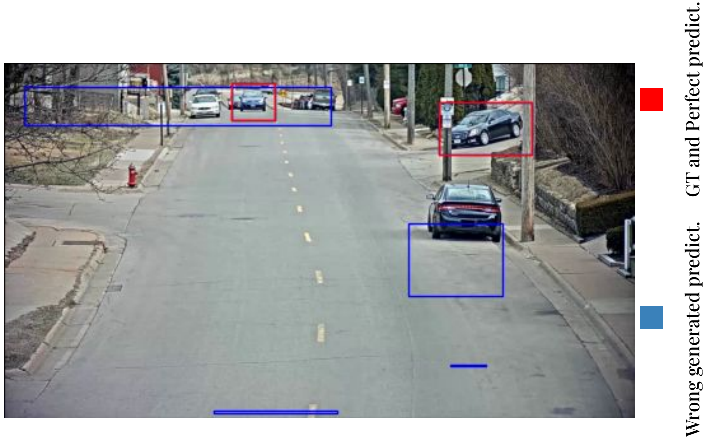
T1.1 IoU & mAP for (Team 3) - [1/3]

- For every BB from the ground truth we can add noise to the position or size of the ground truth and also introduced random generation or elimination of BBs. We use the noises separately.

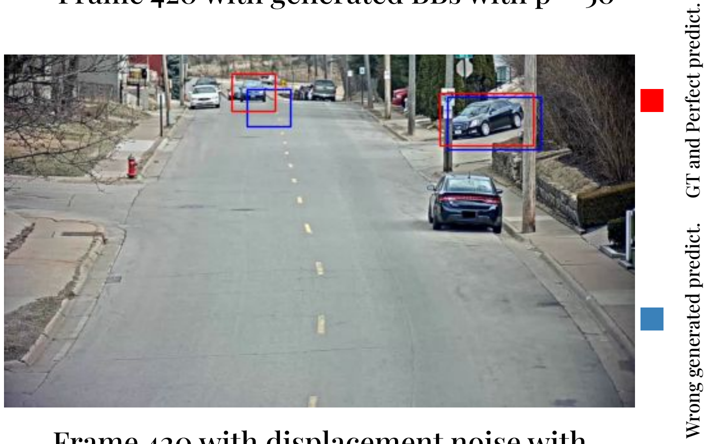
- For every type of noise we have a parameter that we can modify. In case of the position and size noises is the maximum distortion in pixels it produces to the left and top or the width and height respectively.

- When generating BBs, we use a probability for creating random BBs. We can create up to n (in our studies is n=5) BBs with a probability of p for every BB in the ground truth.

- For every BB we have a probability of p to delete it and not consider it as a prediction.

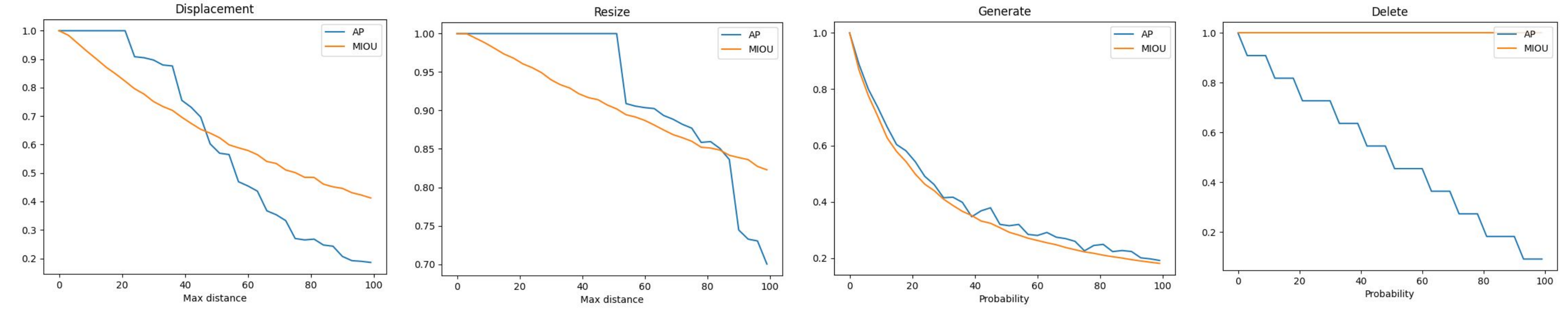


Frame 420 with generated BBs with p = 50

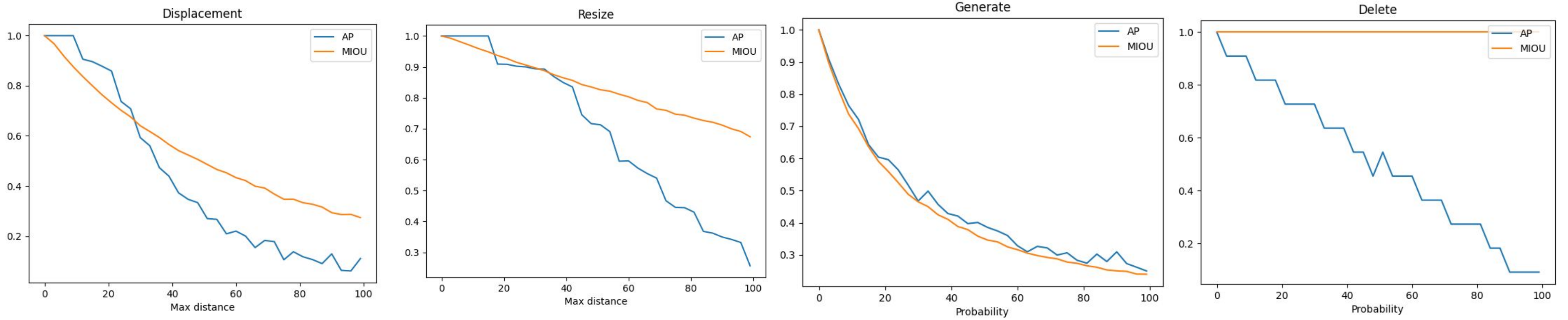


Frame 420 with displacement noise with displacement = 50

T1.1 IoU & mAP for (Team 3) - [2/3]



Equally probable distribution



Gaussian distribution with std max distance/probability

T1.1 IoU & mAP for (Team 3) - [3/3]

- We can see that MIOU **decreases gradually** in all types of noise **except** with the **delete** one. This is due to the fact that we can't compute the IoU between a gt and a prediction that doesn't exist.

- AP is **robust** to small changes in displacement and resize because while the IoU is greater than the threshold, the predicted BB is not discarded as false negative.

- When generating new BBs, as we don't have confidence, we have to create n random ranks and average the AP across these n runs.

- The **random generation** of BBs is the most harmful noise for the AP because when adding more random BBs, is more probable that we don't rank the correct BB in the first position and therefore, reduce the AP.

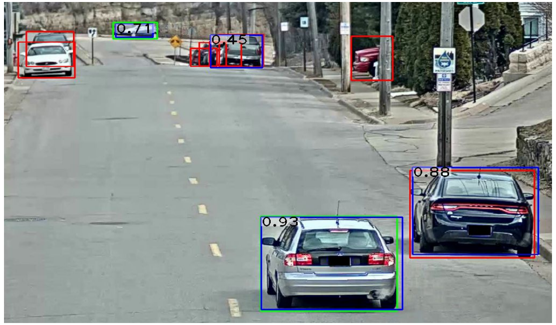
- Whenever we remove a BB from the prediction, the **recall decreases** along with the **AP**.

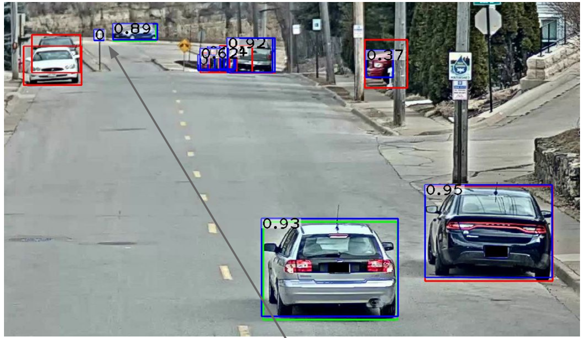
- When using a **Gaussian distribution**, AP decreases faster in displacement and resize noise. This is because the max distribution is now the standard deviation and we have around a 95% to fall inside two standard deviations (e.g. if we have a max distribution of 25, we have a 95% to fall between -50 and 50).

T1.1 IoU & mAP for (Team 4) - [1/3]

- Groundtruth (moving cars)
- Groundtruth (parked cars)
- Predictions

Trying to explain the differences in IoU between the different neural network predictions given we observe that





SSD

We can observe that SSD gives less predictions than YOLO V3 and MASK RCNN, but they seem to be more accurate in terms of IoU.

YOLO V3

YOLO v3 seems to give more predictions, but it tries to create a really small bounding box on some cars, which give a small IoU.

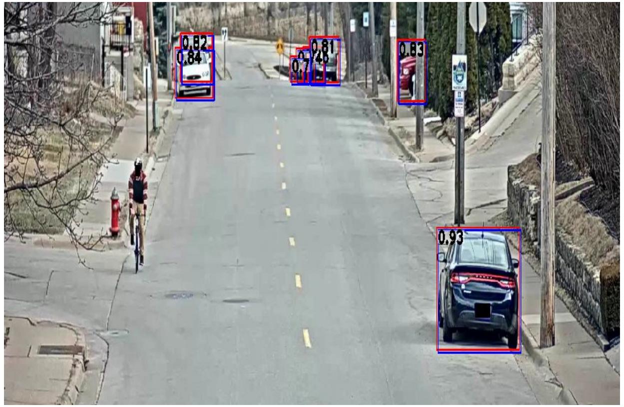
MASK RCNN

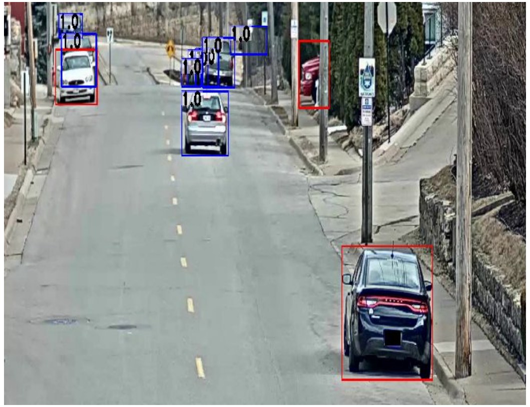
By contrast to YOLO v3, instead of trying to predict the small bounding boxes on cars, the mask rcnn neural network incorrectly predict traffic signals as cars.

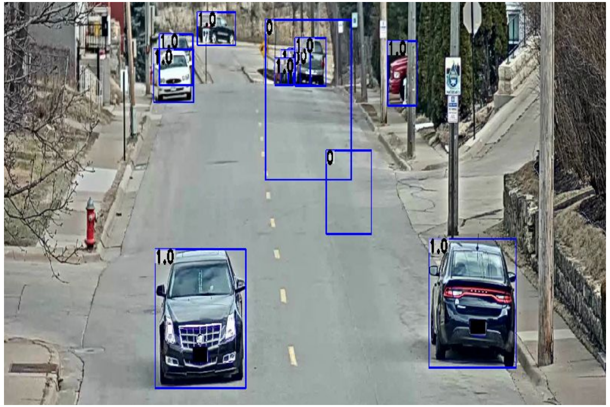
T1.1 IoU & mAP for (Team 4) - [2/3]

Mean	Standard deviation	dropout threshold	generate bbox threshold	mIoU	mAP
0	0	0	1	1	1
4 (pixels)	0	0	1	0.81	0.91
0	4 (pixels)	0	1	0.81	0.91
0	0	0.3	1	0.76	0.64
0	0	0	0.5	1	0.91

T1.1 IoU & mAP for (Team 4) - [3/3]







Modifying mean and std parameters decreases IoU, which lowers mean IoU. It also decreases mAP as in certain cases IoU could be lower than 0.5, which is a FP

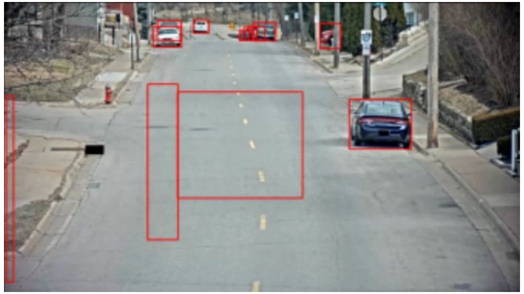
Increasing the probability of images being dropped out decreases mean IoU because in GT images we have IoU=0, and decreases mAP as recall (more FN) decreases

Adding random images decreases mean IoU because in this new images we have IoU=0, and decreases mAP as precision (more FP) increases

T1.1 IoU & mAP for (Team 5) - [1/3]

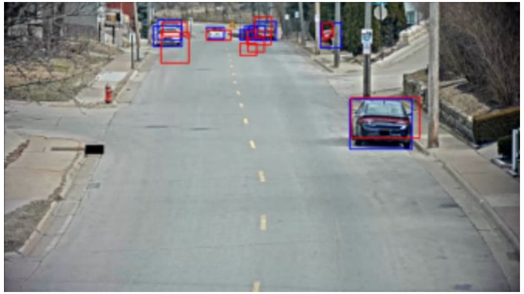
Noise inserted to the annotations:

- Ground truth box
- Detected (annotations with noise) box



Box random generation

Box random removal



Box size and position random noise

T1.1 IoU & mAP for (Team 5) - [2/3]

Size and position mean (pixels)	Size and position standard deviation (pixels)	Box removal probability	Box generation probability	mIoU	mAP
0	0	0	0	1	1
0	0	**0.2**	0	**0.81**	**0.73**
0	0	0	**0.2**	0.91	**0.85**
5	0	0	0	**0.68**	0.91
0	**5**	0	0	**0.78**	0.89
9	**9**	0	0	0.49	**0.34**

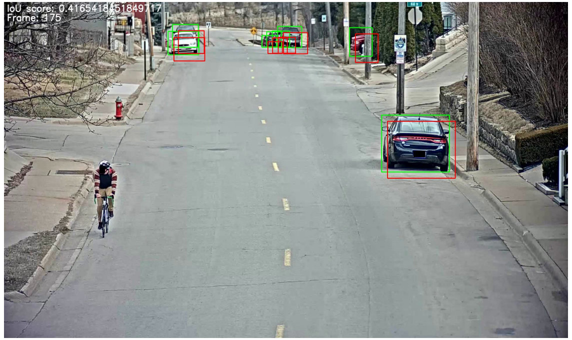
T1.1 IoU & mAP for (Team 5) - [3/3]

- Randomly removing boxes leads to a reduction in both mIoU and mAP due to the introduction of false negatives.
- The addition of random boxes results in a decrease in mIoU and mAP, as it introduces false positives. However, since new boxes can also intersect with objects, the results are better than simply removing boxes.
- Small variations in the position and size of boxes (mean and standard deviation) have a greater impact on mIoU than on mAP. This is because the computation of mAP uses a 0.5 IoU threshold.
- Larger variations in the position and size of boxes result in IoU less than 0.5 and have a significant impact on mAP.

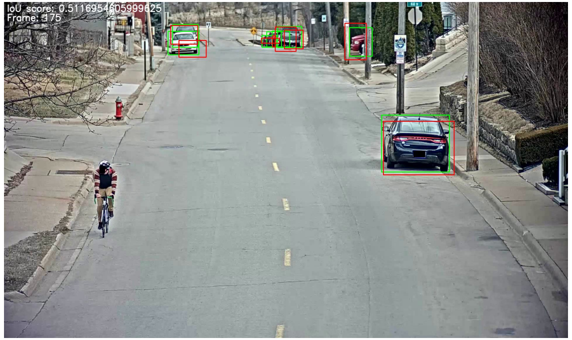
T1.1 IoU & mAP for (Team 6) - [1/3]

- To generate the noisy ground truth annotations, four parameters were considered. The probability of generating random bounding boxes, the probability of removing a bounding box, in addition to the parameters that define the normal distribution, mean and standard deviation.
- When the probability obtained for a bounding box is greater than the probability of being dropped, it will be modified by adding the noise corresponding to the normal distribution defined, otherwise it will be dropped. On the other hand, if the probability is lower than the probability to generate, a random bounding box will be generated in that frame.
- This figure shows the effect of each of the parameters separately.

Mean = 20

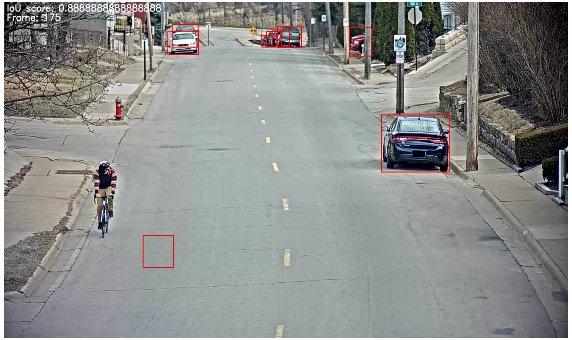


Std = 15



Delete = 0.7

Generate = 0.7



Ground truth Predicted

T1.1 IoU & mAP for (Team 6) - [2/3]

Experiment	mean	std	delete_probability	generate_probability	mIOU	mAP
0	-2	1	0.1	0.1	0.830711	0.810786
1	0	1	0.1	0.1	0.886393	0.902075
2	2	1	0.1	0.1	0.832345	0.899158
3	0	2	0.1	0.1	0.850911	0.90088
4	0	5	0.1	0.1	0.752073	0.808429
5	0	10	0.1	0.1	0.62828	0.630966
6	0	15	0.1	0.1	0.368466	0.367815
7	0	1	0.7	0.1	0.368466	0.355724
8	0	1	0.1	0.7	0.88211	0.767342
9	-2	15	0.1	0.1	0.533034	0.370982

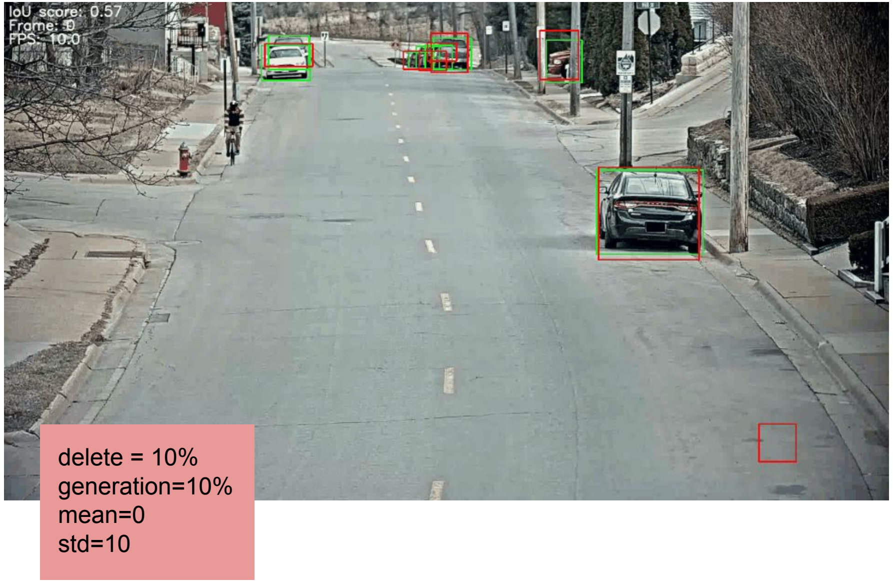
The results show that when we add modifications to the position of the bounding boxes, both metrics decreases slightly.
On the other hand, a larger decrease is obtained by increasing the std.

As we can see in the last experiment, if we modify both, the mean and standard deviation values, we obtain lower values of mIOU and mAP, since the noise is higher.

When we increase the probability of dropping bounding boxes we obtain a lower mAP value. This is due to the penalty of false negatives.

T1.1 IoU & mAP for (Team 6) - [3/3]

Noisy



Ground truth

Predicted

Conclusions

- When we add noise to the position of the bounding boxes, both scores decrease slightly. Depending on the mean of the normal distribution the mAP score decreases more than the mIOU.
- When we add noise in the size of the bounding boxes, both scores decrease considerably. In this case the MAP is more affected as we increase the standard deviation.
- When we eliminate bounding boxes we obtain a higher mIOU compared to the mAP, since, in this case we have more bounding boxes ground truth.
- On the other hand, when we generate more bounding boxes than ground truth, the mIOU decreases, since the randomly generated ones would penalize.

T1.1 Instructors’ feedback

	feedback
Team 1	Nice README.md in github. Code clear and easy to follow (your implementation of IoU?) mAP implementation? How do you scale by confidence in the ground truth? scalings, translations, dropout and adding. No combination. Maybe the changes are too high? Are they random? Nice discussion on mIoU and mAP results on the different perturbations.
Team 2	Nice README.md in github. Link to week1 in the main README.md Code clear and easy to follow (your implementation of IoU?) detectron2 mAP implementation. scale, translation, dropout, missing explanation of “prob.similar” vs “std similar”? No combination. Nice discussions of mIoU and mAP changes (mAP seems more sensible and non-linear). Nice to show qualitative results on one frame.
Team 3	Extremely well documented on the README.md (good job!) mAP implementation? scalings, translations, dropout and adding. No combination. what is the difference between the first and second row on slide 2? Did you take into account FP? Nice discussion in slide 3.

T1.1 Instructors' feedback

	feedback
Team 4	Good analysis of the different detectors Text in slide 17 lacks clarity. There are errors such as using the word 'images' instead of 'detections' No combinations of different types of noise No README.md in github for this week. It should explain how to replicate your results
Team 5	Added boxes should be similar in size to the existing GT boxes. In the example given they semm much bigget. Not clear in the text if the added boxes are generated with random size. No combinations of different types of noise README.md for the week should explain how to replicate your results
Team 6	Good explanations on how the noise is added. Examples in the figure in slide 21 should be explained better Very good week1 README.md in week1.

T1.2 mAP for provided object detections

- Compute the mAP for the provided detections (mask_rcnn, ssd512, yolo3)

T1.2 mAP for provided object detections

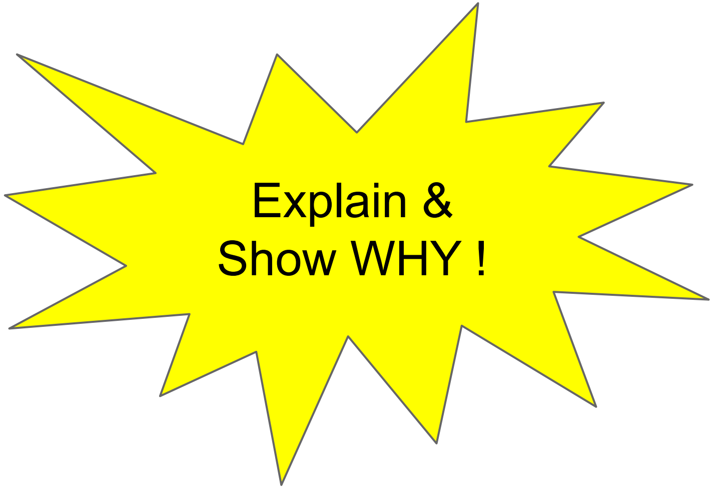
Team	Mask RCNN	SSD512	Yolo3	implementation
1	0.4378	0.3808	0.4339	Pascal VOC
2	0.4114	0.3629	0.4322	Pascal VOC
3	0.4112	0.3625	0.4321	Pascal VOC
4	0.4128	0.3631	0.4427	Pascal VOC
5	0.4122	0.3631	0.4394	Pascal VOC
6	0.4127	0.3631	0.4428	Pascal VOC

T1.2 Instructors' feedback

	feedback: wow! nearly there!! ;)
[Team 1]()	Small difference with other teams. Do you have any ideas why?
[Team 2]()	
[Team 3]()	
[Team 4]()	
[Team 5]()	
[Team 6]()	

T2 IoU vs time

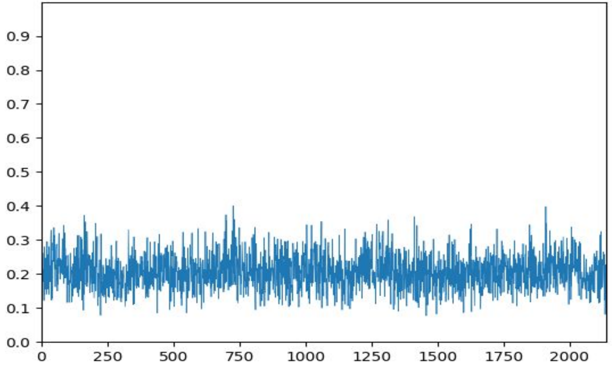
- Temporal analysis of the results
    - Graph: IoU vs #frame



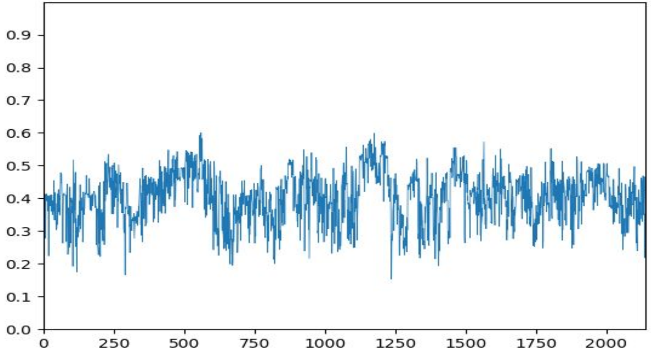
T2 IoU vs time (Team X) - Copy & paste (max 3. pages)

T2 IoU vs time (Team 1) - Results after all frames - [1/3]

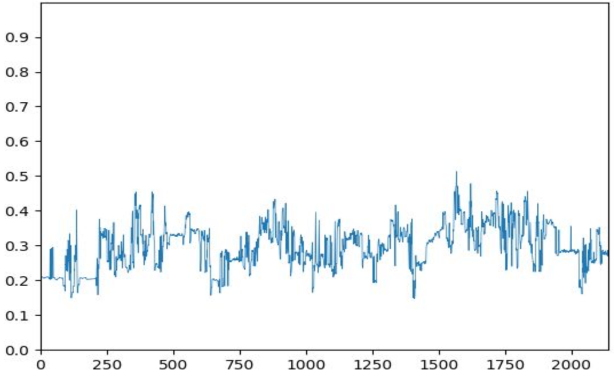
Noisy annotations
mIoU = 0.2063



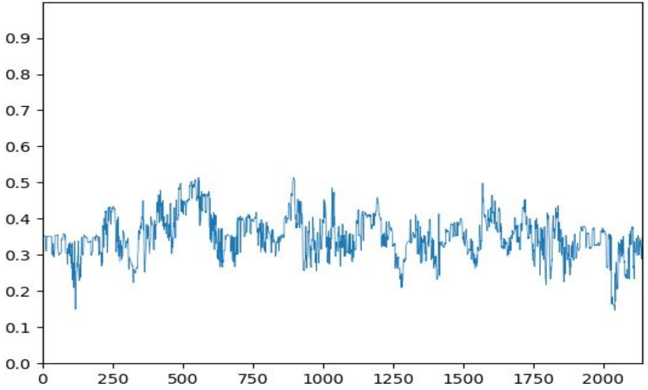
Mask RCNN
mIoU = 0.4035



SSD
mIoU = 0.2963



YOLOv3
mIoU = 0.3511

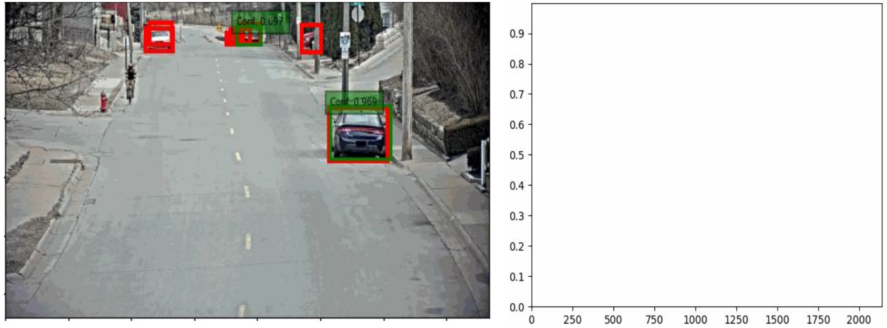


T2 IoU vs time (Team 1) - [2/3]

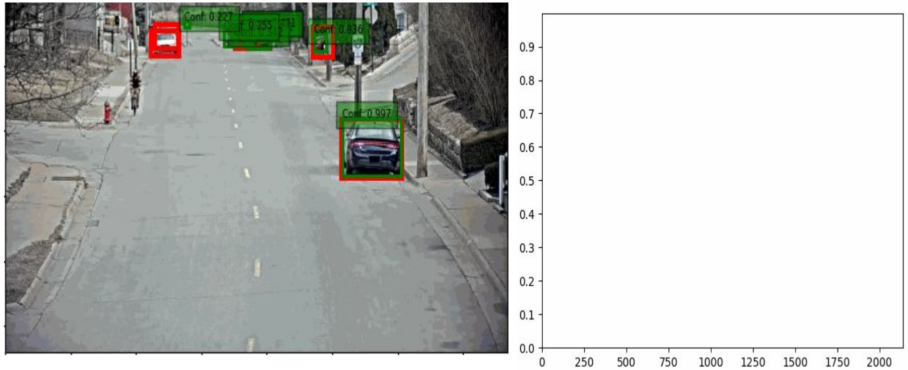
Ground truth box

Detected (annotations with noise) box

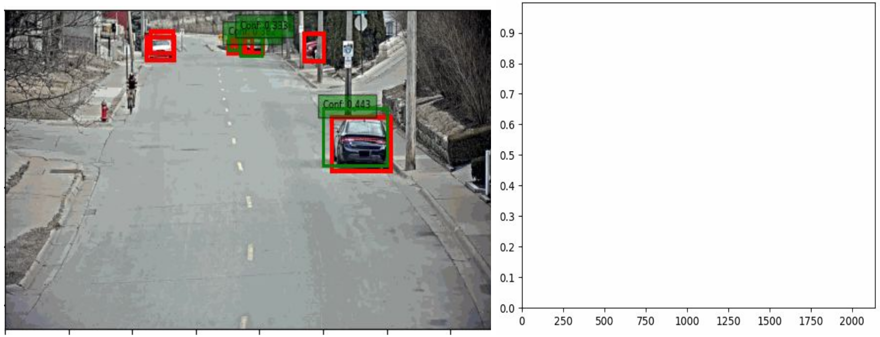
SSD
mIoU = 0.2963



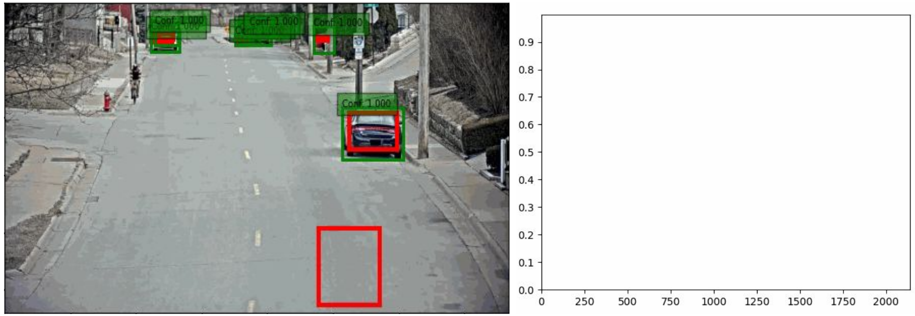
Mask RCNN
mIoU = 0.4035



YOLOv3
mIoU = 0.3511



Noisy annotations
mIoU = 0.2063



T2 IoU vs time (Team 1) - [3/3]

Observations

- The biggest mean Iou for all the sequence is obtained by the maskRCNN model.
- All of the detections from the different models present a noticeable variation in the IoU during the sequences. Nevertheless the ground truth annotations with added noise variation is still larger.
- All the models are capable of detecting cars that are close to the camera with a high confidence but struggle with far situated cars mainly due to a low confidence in the detection.
- In the sequence it can be observed how in the end of the street there are cars appearing and disappearing constantly, as well as from the right of the image

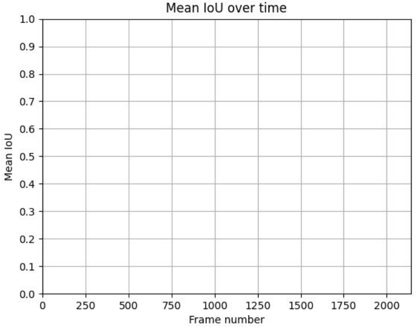
Conclusions

- The cars at the end of the street are hard for the models to detect as they are small and appear and disappear from the scene rapidly. This contributes to the noisy IoU that we see in all the models.
- In some frames there also happen occlusions that makes the models perform worse.
- The highest mean IoU from the maskRCNN model is understandable as it is a two-stage detector, having more parameters than the one-shot detectors, the SSD and the YOLOv3
- Overall, the 3 detectors achieve bad mIoU, especially because of FN that repeat a lot in each frame (background cars almost never detected, having a lot of IoU = 0)

T2 IoU vs time (Team 2) - [1/2]

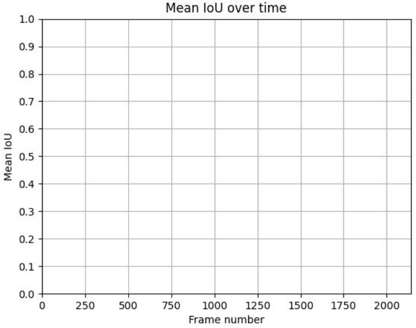
When comparing the annotated video with the different provided detections, we note that **cars in the foreground of the scene are easily detected**, while those in the background or in partially blind spots (caused by object occlusions or the boundaries of the scene) are very hard to detect, and suffer from a blinking and shaking effect on the detection due to the low confidence of the prediction that causes an unstable IoU between frames in all the cases.
Regarding the **IoU, it fluctuates around 0’3 and 0’6 over time for all the models**. And in the case of the baseline noisy predictions, the IoU variation between frames is even higher, reaching from 0’15 to 0‘75, but it gets the highest mean IoU overall. However, among the models, the MaskRCNN detections are the ones with less variations and highest mean IoU.

MaskRCNN



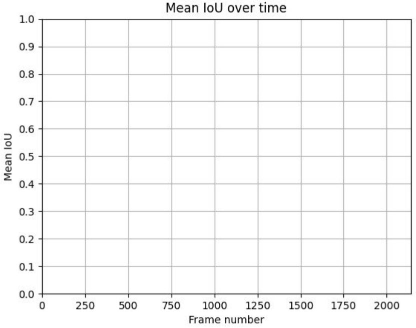
Mean IoU
.5093

SSD512



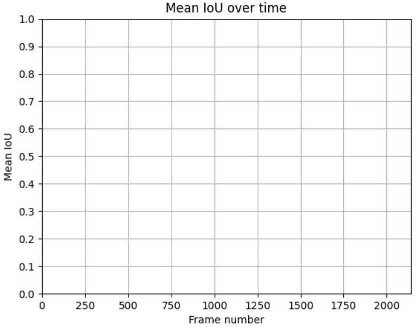
Mean IoU
.3784

YOLO3



Mean IoU
.4491

Noisy annotations

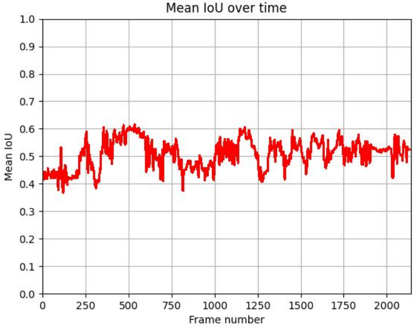


Mean IoU
.5626

T2 IoU vs time (Team 2) - [2/2]

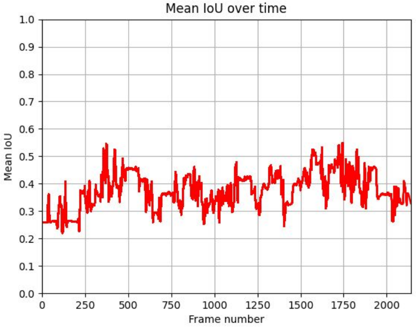
In conclusion, when observing the entire sequence and IoU over time, we obtain the **highest mean IoU with the baseline noisy annotations, i.e. .56 IoU**. However, it is clear that its fluctuation of the IoU along frames is quite extreme and thus, it does not provide reliable nor persistent detections. In fact, this increase in the mean IoU may be due to chance when adding noise to the bounding boxes.
Also, note that SSD512 and YOLO3 models are single-shot detectors, while MaskRCNN is a two-stage architecture (it has first a Region Proposal Network in the feature extraction stage, and a ROI alignment for the detection step in the second stage). Hence, this may be one of the main reasons why **MaskRCNN obtains the highest mean IoU with less fluctuation**. However, SSD512 and YOLO3 perform worst and suffer a slightly highest variation in terms of IoU.

MaskRCNN



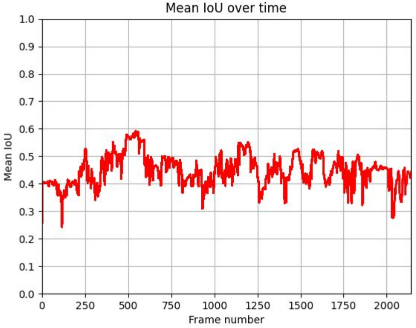
Mean IoU
.5093

SSD512



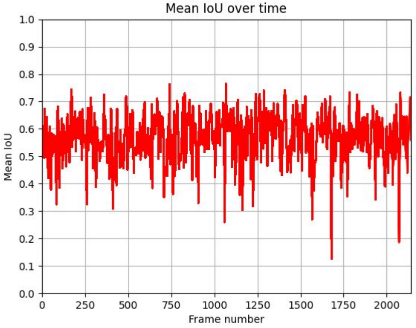
Mean IoU
.3784

YOLO3



Mean IoU
.4491

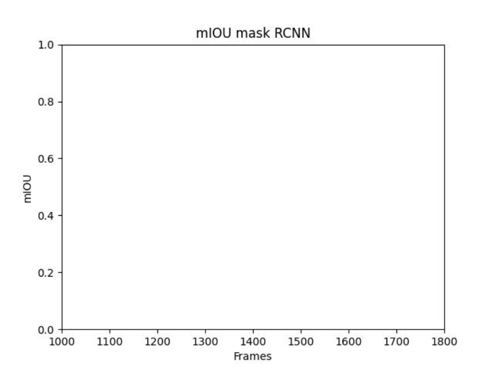
Noisy annotations



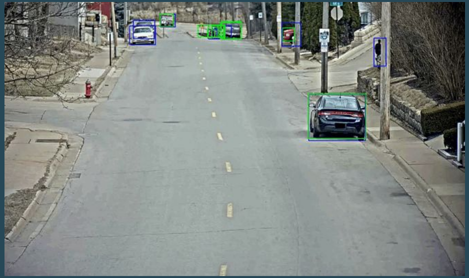
Mean IoU
.5626

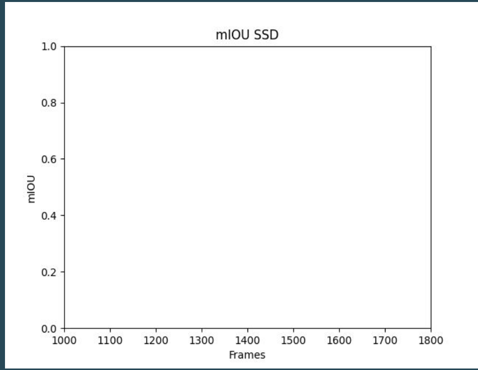
T2 IoU vs time (Team 3) - [1/2]

Detected : \\\\\
Ground Truth: \\\\\

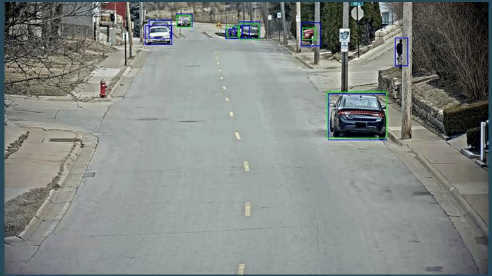


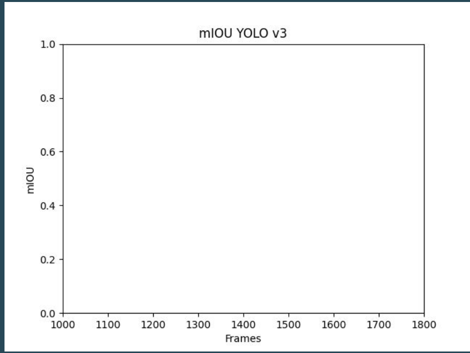
mask RCNN mIOU:0.598
stdIOU:0.213





SSD mIOU:0.768
stdIOU:0.139





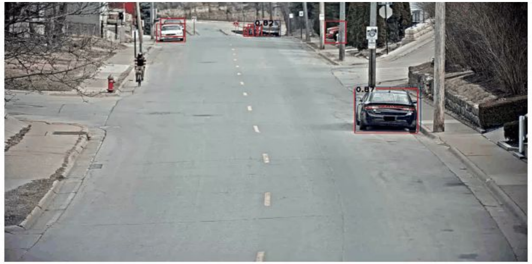
YOLOv3 mIOU:0.459
stdIOU:0.393

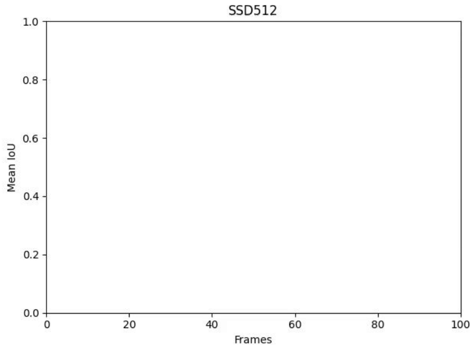
T2 IoU vs time (Team 3) - [2/2]

- Among the detector SSD has the highest mIOU and lest standard deviation and performs better than the other two. Yolo performs the worst in terms of mIOU.
- One reason why the model has performed better than the others may be due to the selected frames the model performs better due to training.
- All the models are able to detect the cars that are very near to the camera correctly and with high confidence.
- And similarly all the models fail to detect cars that are far away from the camera and hence appear very small.(white car parked on the left side of the frame)
- Also due to the motion of the cars, there is occlusion in some cases and hence the performance is hampered during these instances. This can be seen with the rapid drops in some parts of the curves and can be related a car entering or exiting the scene.

T2 IoU vs time (Team 4) - [1/3]

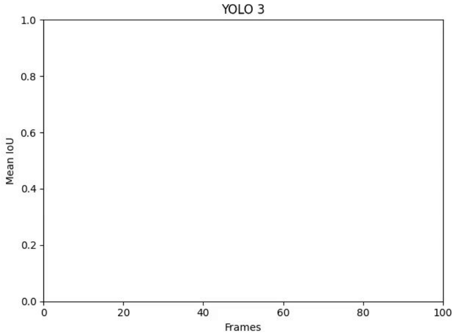
SSD512





We can see that it manages to even capture the car on the background, although there are some instants where it does not manage to detect it. Cars with occlusions are also detected correctly.

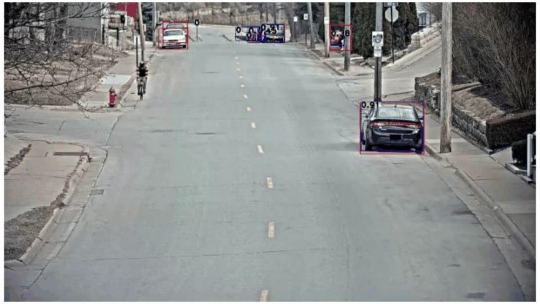
YOLO 3

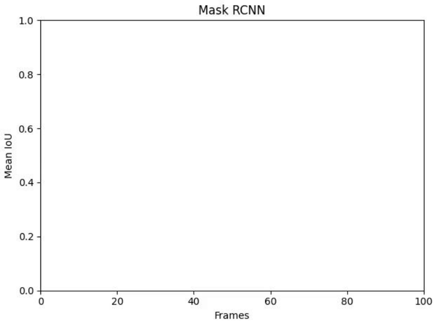


In this case, YOLO detects properly the car on the background during the entire sequence. YOLO's predictions are the most robust compared to the other algorithms.

T2 IoU vs time (Team 4) - [2/3]

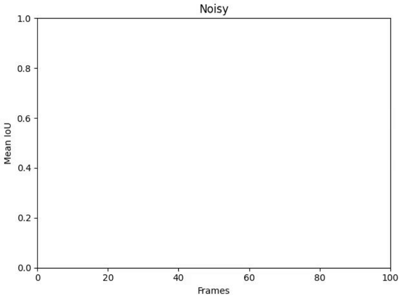
Mask RCNN





Mask RCNN seems to misdetect additional cars on the background, and bounding boxes seem to be more jittery and unstable than the previous approaches.

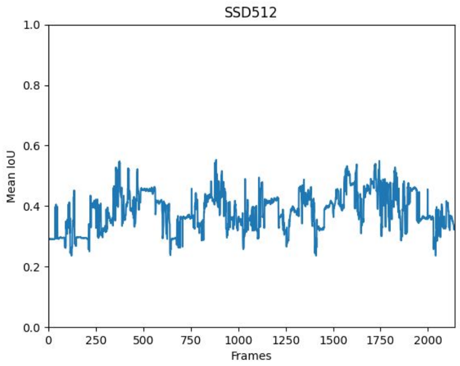
Noisy

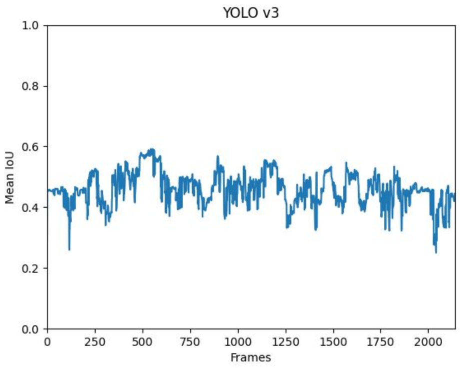


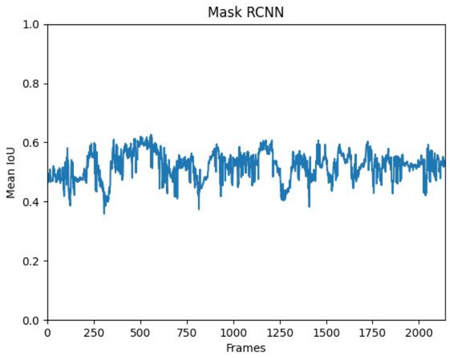
Adding noise on the ground truth (mean=0 std=10, thresh_dropout = 0.3, thresh_boxes = 0.3) decreases the mIoU by around 40%, although it is still closer and better than the rest of the predictions, but more jittery.

T2 IoU vs time (Team 4) - [3/3]

As a result of the previous slide explanations, given that YOLO v3 and MASK RCNN may be able to create more predictions (with some of them being accurate) than SSD, it may result in having a higher mean average precision (mAP).







mAP SSD: 0.3631

mIoU SSD: 0.3831

mAP YOLO V3: 0.4427

mIoU YOLO V3: 0.4585

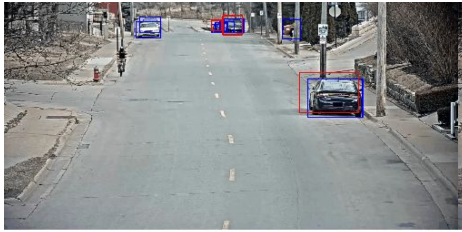
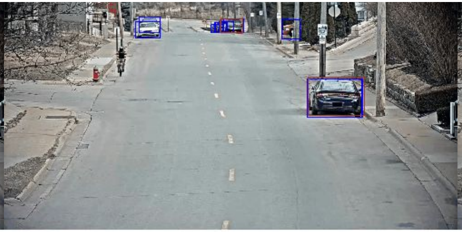
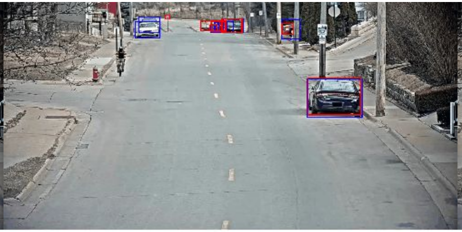
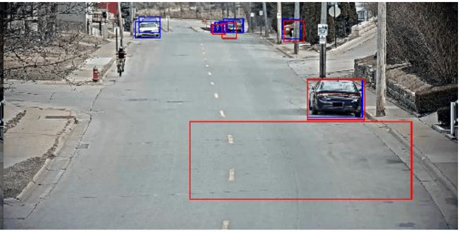
mAP MASK RCNN: 0.4128

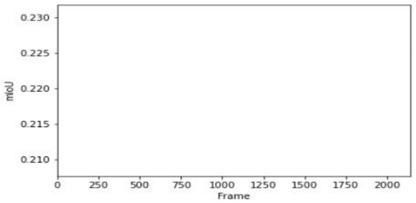
mIoU MASK RCNN: 0.5192

T2 IoU vs time (Team 5) - [1/2]

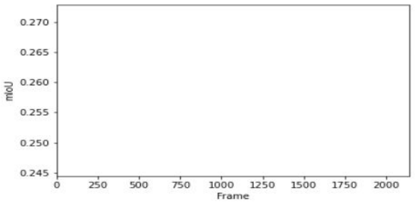
- Ground truth box (blue)
- Detected box (red)

YOLOV3	SSD512	MASK RCNN	NOISY ANNOT. (mean: 5, std: 5, rem_p: 0.2, gen_p: 0.2)

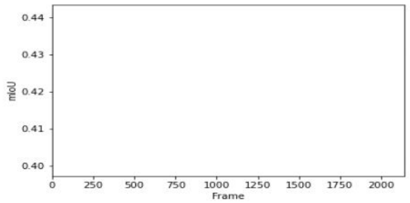







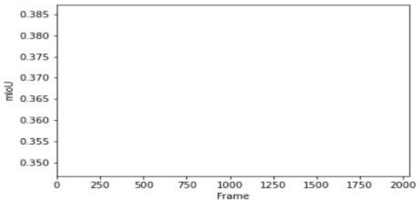
mIoU: 0.39



mIoU: 0.37



mIoU: 0.49



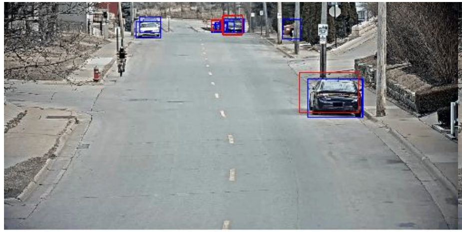
mIoU: 0.51

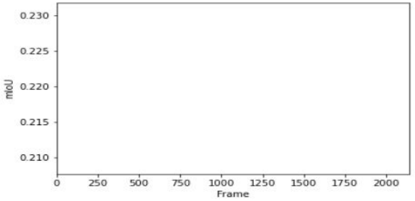
T2 IoU vs time (Team 5) - [2/2]

■ Ground truth box

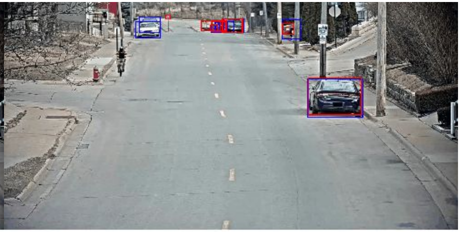
■ Detected box

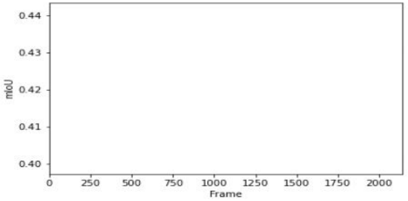
YOLOV3





MASK RCNN



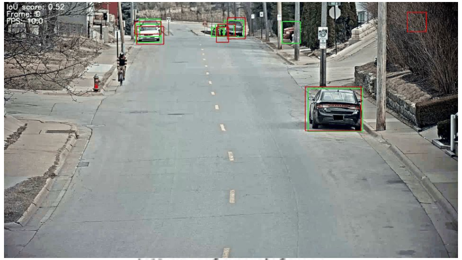


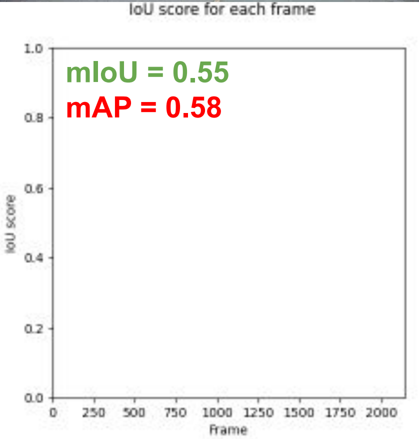
- The best overall results were obtained using Mask RCNN's detections. This could be due to more accurate bounding box prediction, even if it predicts fewer objects (resulting in a lower mAP value than Yolo3).
- It can be observed that better mIoU values are obtained when there are larger objects (cars) closer to the camera, as they are easier to detect accurately.
- In every network's detections, there are some cars (located in the top left corner) that are never detected, which negatively affect the results.
- The Faster RCNN model struggles with detecting objects that are partially occluded or have unusual shapes, such as bicycles(or pedestrians), as an observation, the MIoU values for these classes can be lower than for cars, which are more easily distinguished by tjeri size and shape.

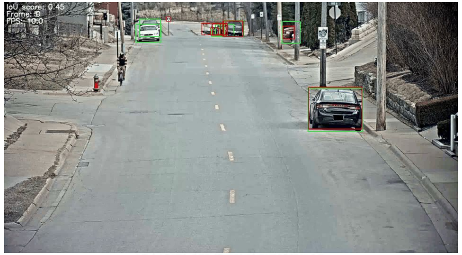
T2 IoU vs time (Team 6) - [1/3]

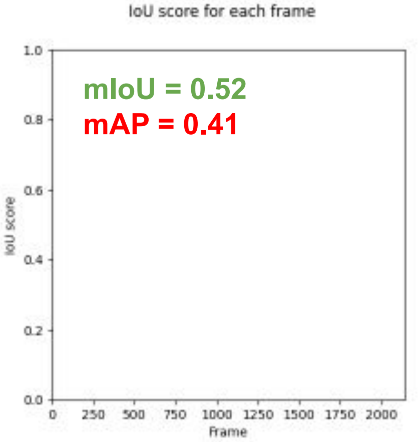
(*)dropout = 60% // generation=10% // mean=0 // std=10

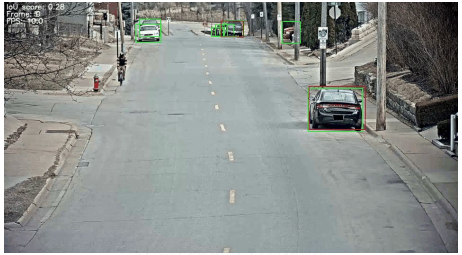
NOISY(*)	MASK RCNN	SSD512	YOLOV3

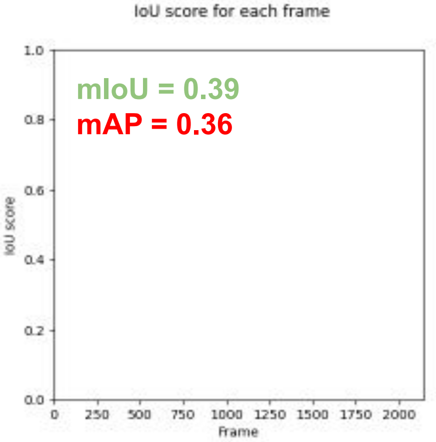


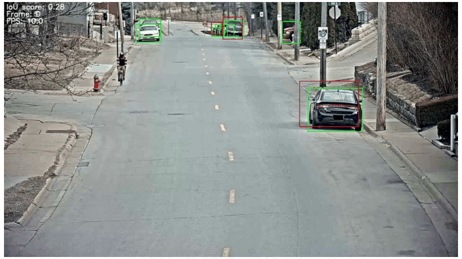


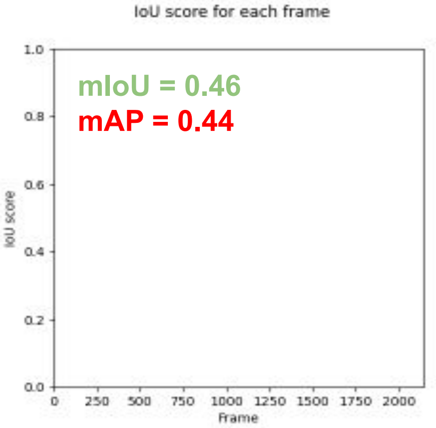










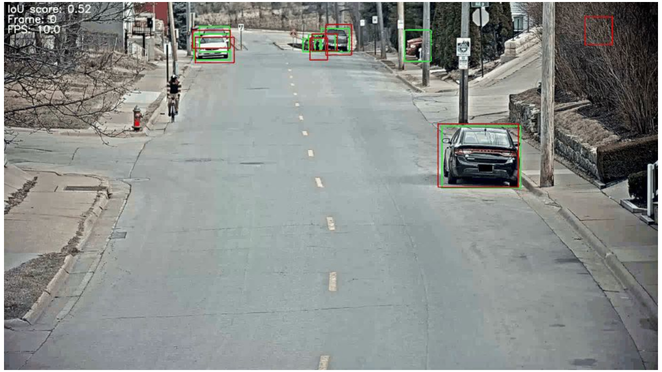


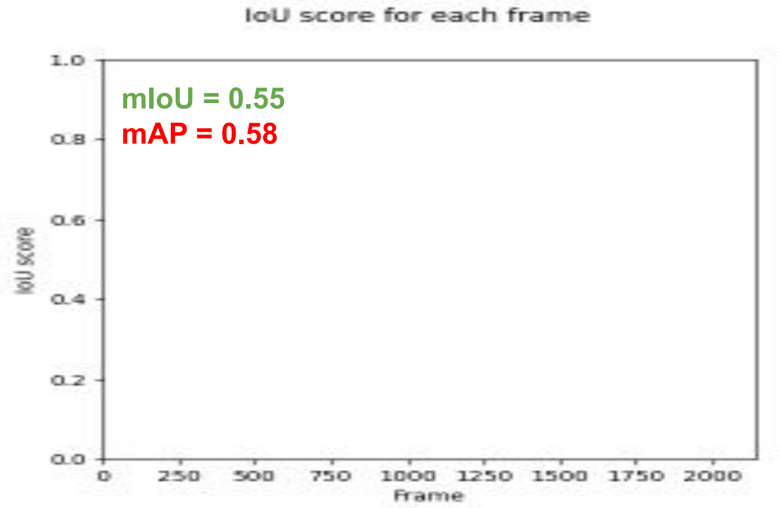
- As far as IOU is concerned, we can observe that mask-rcnn and YOLO seem to perform slightly better than SSD.
- *Cars that are close are better detected than those that are far away.*

Ground truth   Predicted

T2 IoU vs time (Team 6) - [2/3]

(*)dropout = 60% // generation=10% // mean=0 // std=10



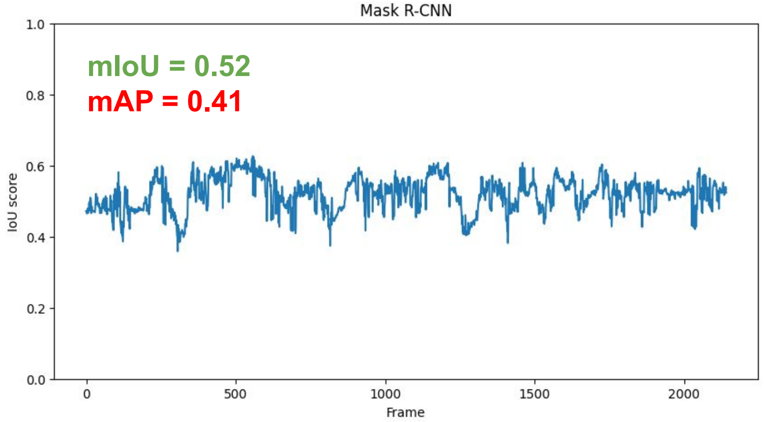


NOISY(*)

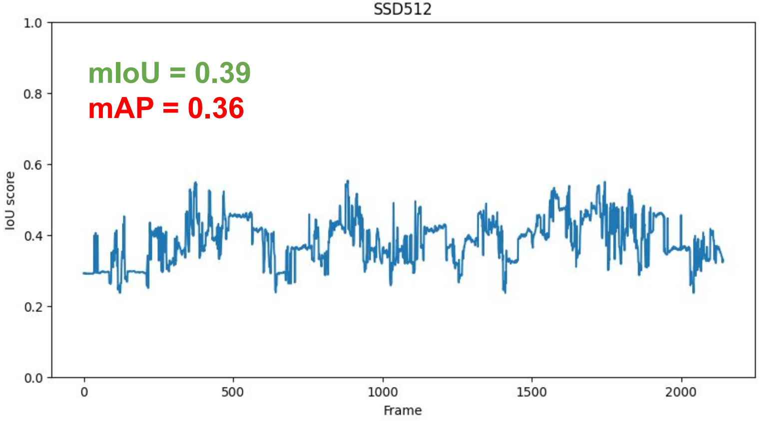
- Note that the plot is very noisy due to the random generation of the data. We used a high value for standard deviation, which resulted in high variance between 0.3 and 0.7 approximately.
- As expected there is some variance in the results due to the motion of the cars and the distance from the camera. H
- Noisy case reach the best mIoU score but does not produce consistent detections over time.

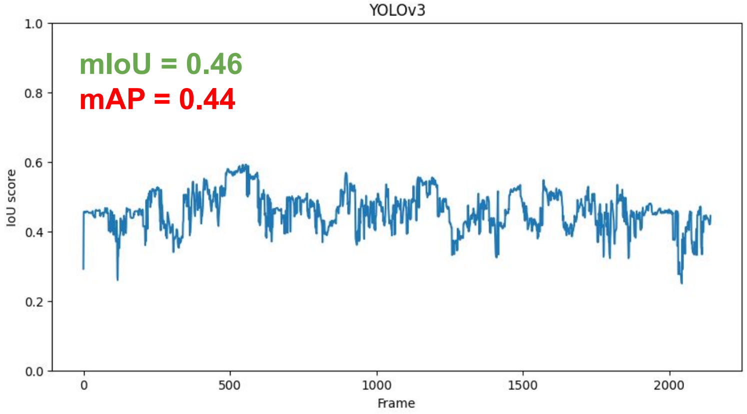
Ground truth   Predicted

T2 IoU vs time (Team 6) - [3/3]



- Based on the plots and mean IoU (mIoU) scores, the Mask R-CNN model outperforms the YOLOV3 and SSD512 models in terms of accurately detecting objects.
- It's important to note that there is some inherent variation in the IoU scores due to the motion of the cars, as discussed in the previous slide.
- One common challenge that all three models struggle with is detecting parked cars in the distance. This may be due to the lower resolution or smaller size of the objects.
- Another common problem is related to the partial visualizations, when objects enter or leave the scene. This can result in incomplete or partial detections, which affects the overall performance of the models





T2 Instructors' feedback

	feedback
Team 1	Both qualitative and quantitative results are shown. Synchronized graph/images Conclusion → mask-rcnn best All detectors struggle with far away cars (small objects)
Team 2	Both qualitative and quantitative results are shown. Synchronized graph/images Conclusion → mask-rcnn best All detectors struggle with far away cars (small objects) Why is the mIoU curve in red in slide 2? Label everything, green boxes are GT or detections?
Team 3	Both qualitative and quantitative results are shown. Synchronized graph/images Bonus points for showing std. Conclusion → SSD best (0.7 mIoU), why? Bad behaviour on occlusion.

T2 Instructors' feedback

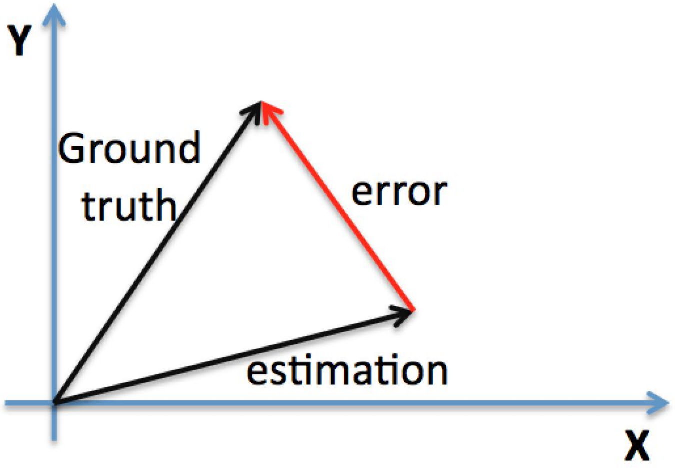
	feedback
Team 4	Both qualitative and quantitative results are shown. Synchronized graph/images Conclusion → maskrcnn and yolo best. Lacks discussion on best miou
Team 5	Both qualitative and quantitative results are shown. Synchronized graph/imagesConclusion → mask-rcnn best. Discussion in terms of IoU and mAP. Good!
Team 6	Both qualitative and quantitative results are shown. Synchronized graph/imagesConclusion → mask-rcnn best. No analysis in terms of mAP Problems on partial visualizations

T3.1 MSEN & T3.2 PEPN

- **MSEN:** Mean Square Error in Non-occluded areas
- **PEPN:** Percentage of Erroneous Pixels in Non-occluded areas

Consider only non-occluded areas (NOC). Indicated by *flow_noc* in the data.

Consider erroneous those pixels whose motion vector error is > 3.



T3.1 MSEN & T3.2 PEPN (all teams fill table)

	MSEN		PEPN	
Team	**Seq 45**	**Seq 157**	**Seq 45**	**Seq 157**
1	10.6271	2.7504	78.5603%	34.0476%
2	10.63	2.75	78.56%	34.05%
3	10.62707	2.7503	78.560%	34.04758%
4	10.6271	2.7504	78.5603%	34.0476%
5	10.627	2.75	78.56%	34.0476%
6	10.6271	2.7504	78.5603%	34.0476%

T3.1 & T3.2 Instructors' feedback

All values equal!!!

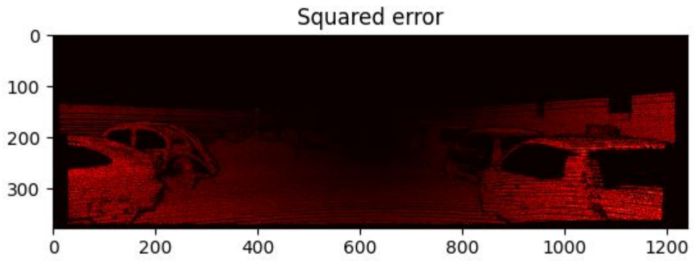
	feedback
Team 1	
Team 2	
Team 3	
Team 4	
Team 5	
Team 6	

T3.3 Analysis & Visualizations

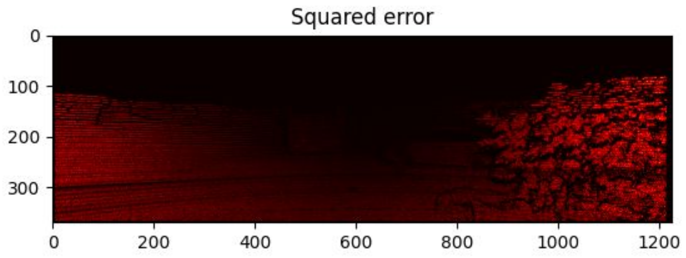
Discuss the obtained results and generate visualizations that help understanding them.

T3.3 Analysis (Team X) - Copy & paste (max 3. pages)

T3.3 Analysis (Team 1) - [1/3]



Squared error for sequence 000045



Squared error for sequence 000157

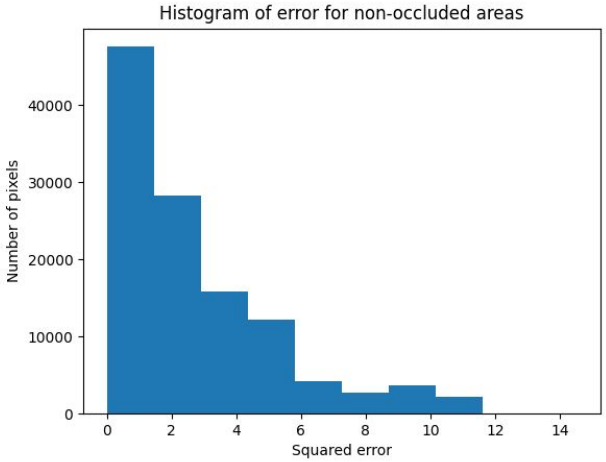
Redish pixels indicate where we have larger errors in terms of squared error.

The more red the pixel the higher the difference between the ground truth and the predicted optical flow.

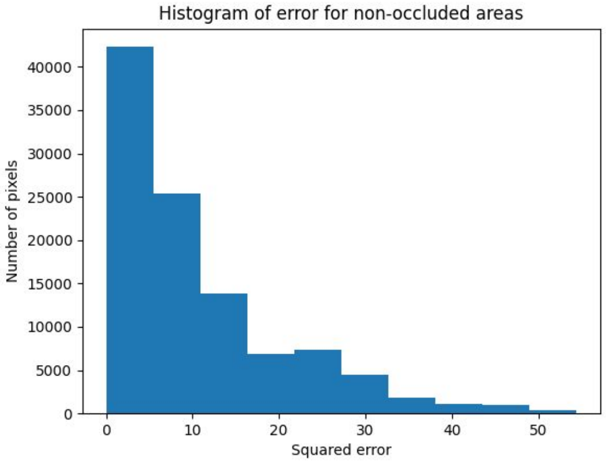
On the other hand, black pixels indicate that the motion of the objects in the sequence have been accurately predicted.

T3.3 Analysis (Team 1) - [2/3]

Both histograms errors show that in both sequences the majority of pixels have very low squared error. As it was already seen in the previous images.



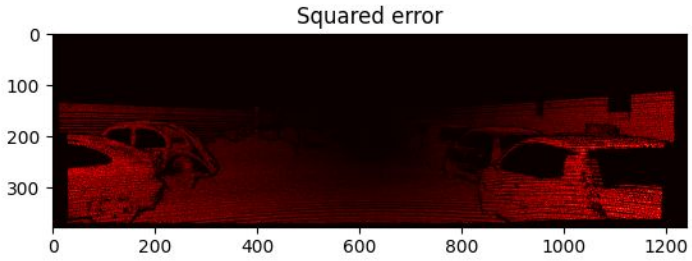
Histogram error for sequence 000157



Histogram error for sequence 000045

T3.3 Analysis (Team 1) - [3/3]

From the visual results we can conclude that the given predictions were good at estimating the optical flow of those points far from the camera, and as a result did not have much motion. But struggled at predicting the optical flow of those points closer to the camera and the motion was more noticeable.

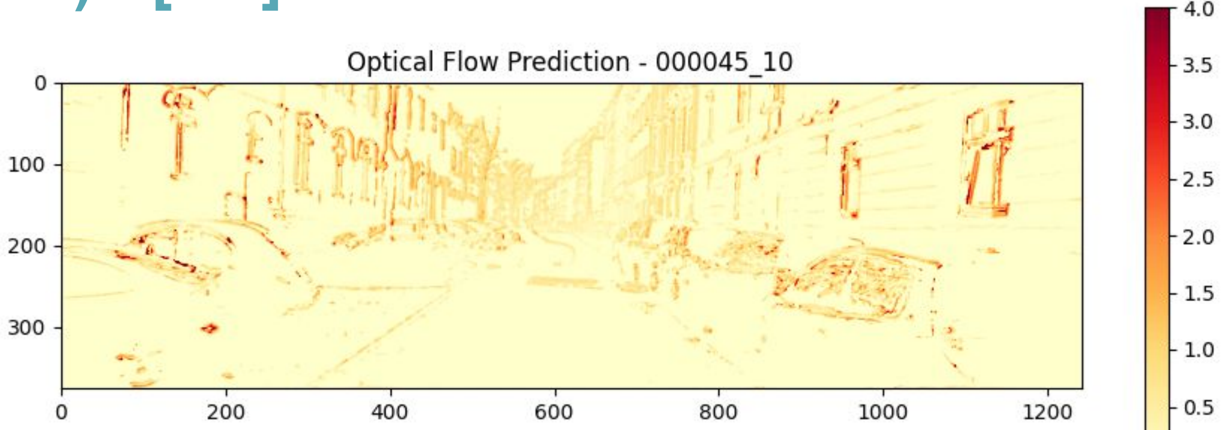


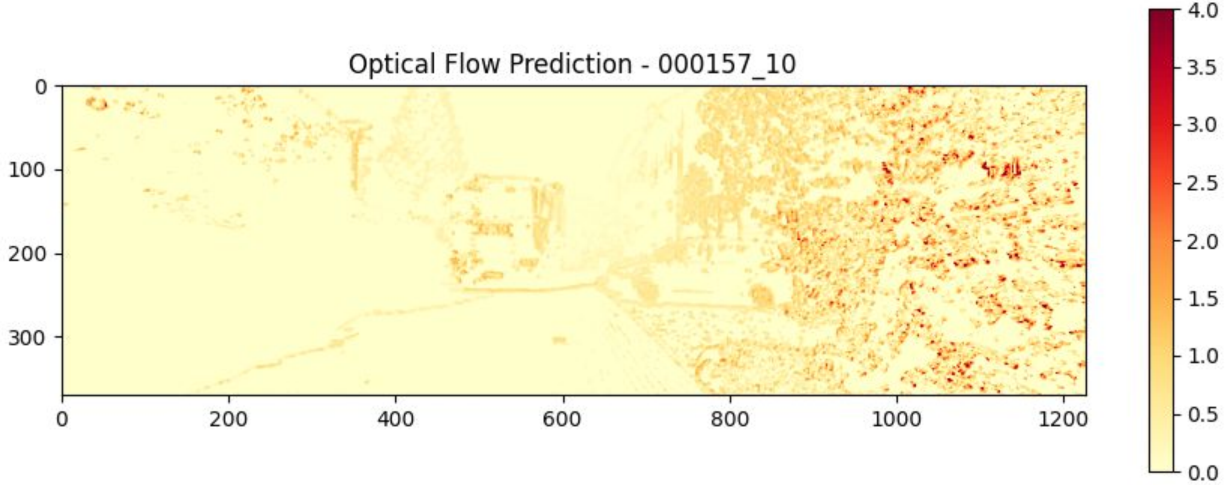
T3.3 Analysis (Team 2) - [1/3]

We plot the **magnitude** of the **predicted** optical flow. We can readily notice that it is higher on distinctive points, namely corners.

Note how on **textureless areas** and to a lesser extent also **edges**, such as the road the building façades, or even the car windows, the predicted optical flow has very low magnitude.

This is accounted for by the fact that the Lucas-Kanade method utilizes the **Hessian** matrix, a.k.a the **Structure Tensor**, which is essentially the same matrix used in **Harris' corner** detection algorithm. Flat regions and edges are not taken into account by the Structure Tensor.





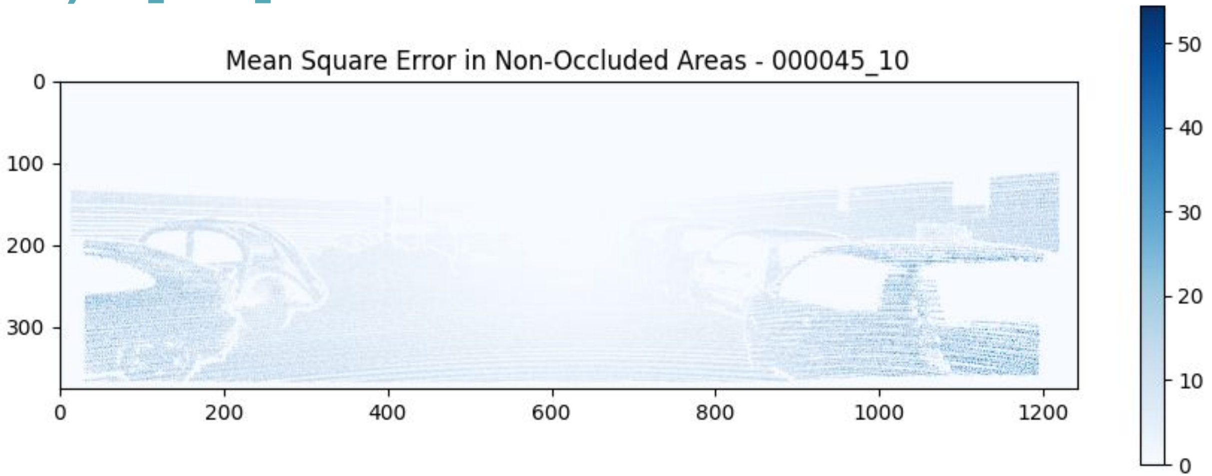
T3.3 Analysis (Team 2) - [2/3]

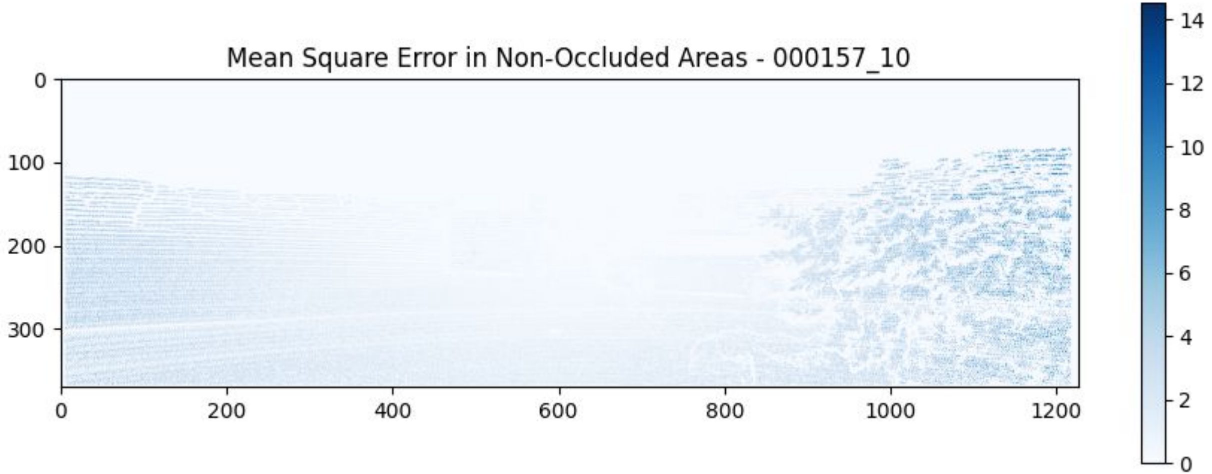
The MSE is lower on the road, i.e. on the areas that lie along the moving direction (assuming the car is moving ahead).

The MSE is higher on surfaces whose corresponding viewing direction from the camera center is at a high angle w.r.t. the moving direction. Thus, the cars (top) and vegetation (bottom) on the road sides have a higher MSE.

This might be because of the texture of the road and the forward motion of the camera. The road being textureless camouflages the real motion (there is less gradient there, so the method suffers from the aperture problem); and the forward motion of the camera implies that the apparent motion grows with the distance to the pole of expansion. This hinders the motion coherence and the small motion assumptions.

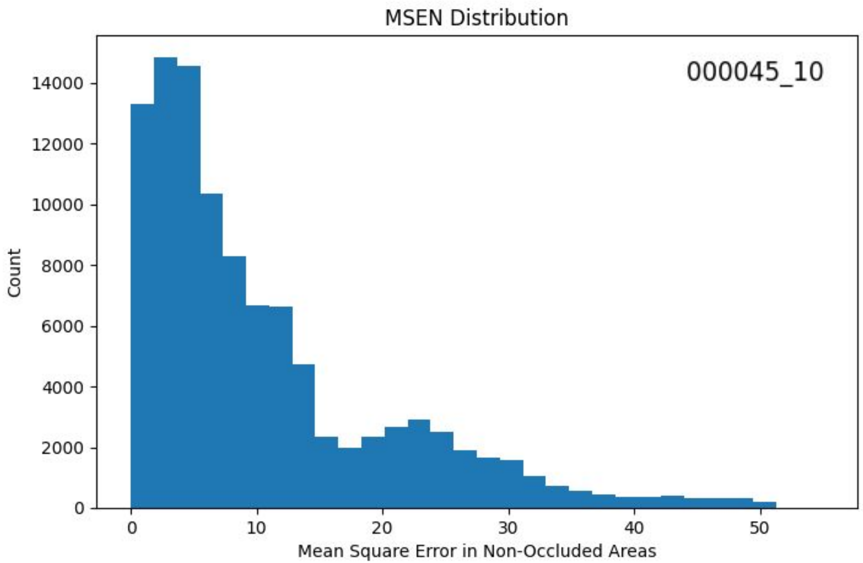
To deal with larger motion on the sides, we could use spatial pyramids.

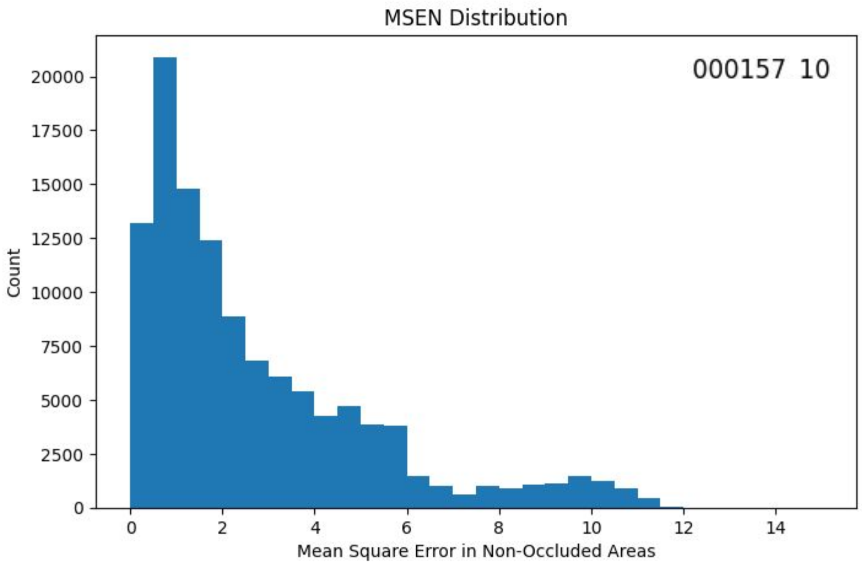




T3.3 Analysis (Team 2) - [3/3]

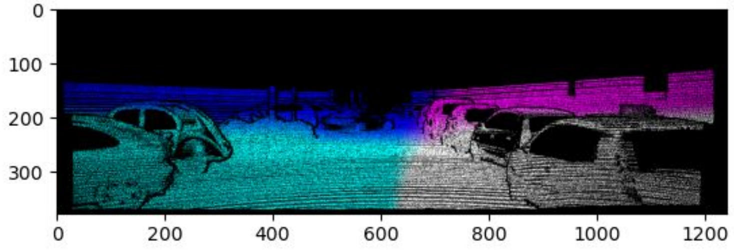
In general, the MSEN values are concentrated on the left mode of the distribution. We can therefore say that overall the results from the Lucas-Kanade algorithm are consistent with the ground truth.
However, the error presents a fat right tail in both of the studied sequences. This means the approach is failing in a significant amount of pixels. Notice the error distributions for both images seem to be bimodal, with a second lobe corresponding to the higher error, which we can associate (based on the MSE plot in the previous slide) to the sides of the road.
It is also worth mentioning that in sequence 000045, the magnitude of the error is in general higher by an order of magnitude, compared to sequence 000157. It seems that the algorithm is struggling with the cars in 000045 more than with the left and right façades of 000157.



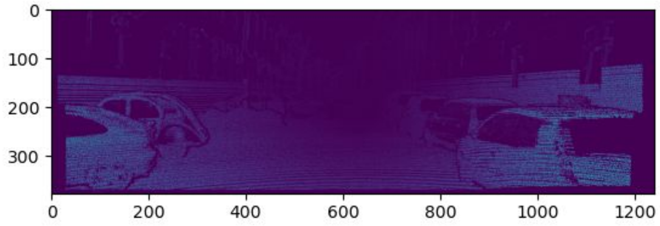


* Pixels with invalid Optical Flow annotations were discarded prior to generating the histograms.

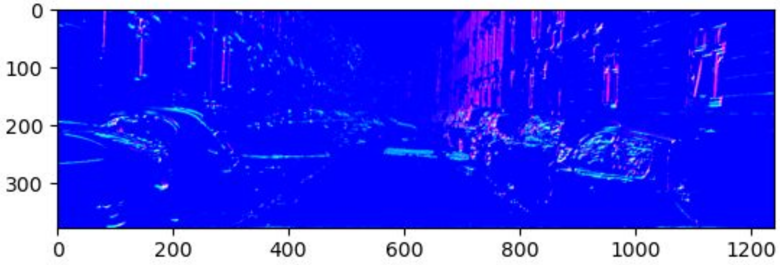
T3.3 Analysis (Team 3) - [1/3]



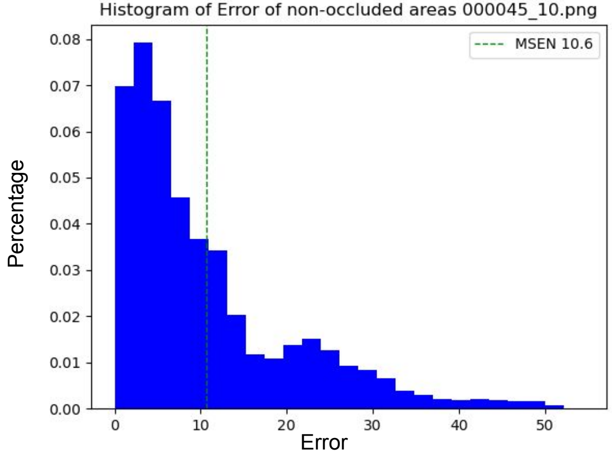
Optical flow ground truth for image 45



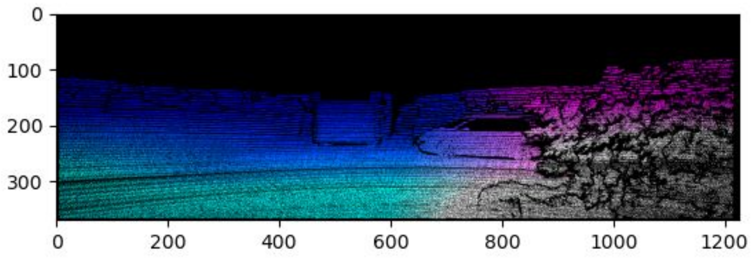
Optical flow error for image 45



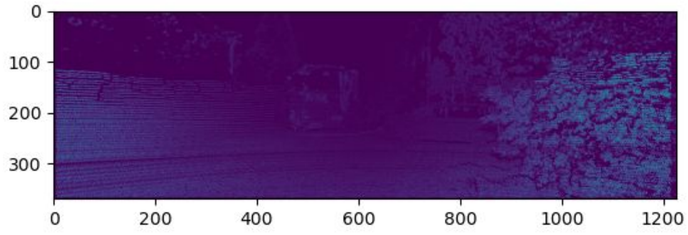
Optical flow for prediction image 45



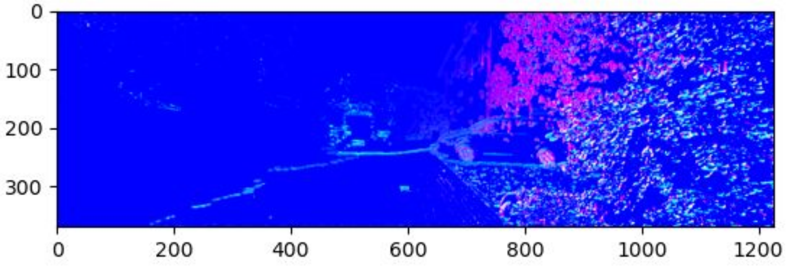
T3.3 Analysis (Team 3) - [2/3]



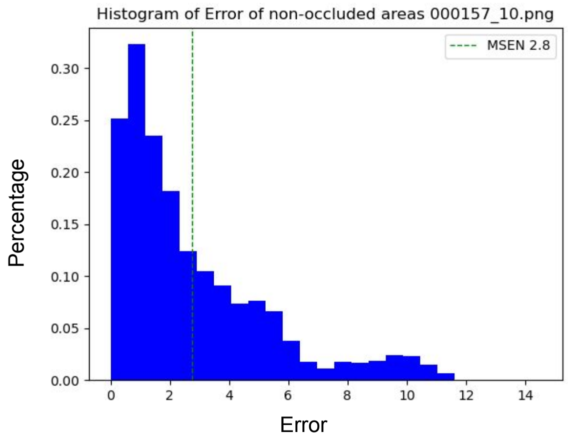
Optical flow ground truth for image 157



Optical flow error for image 157



Optical flow for prediction image 157

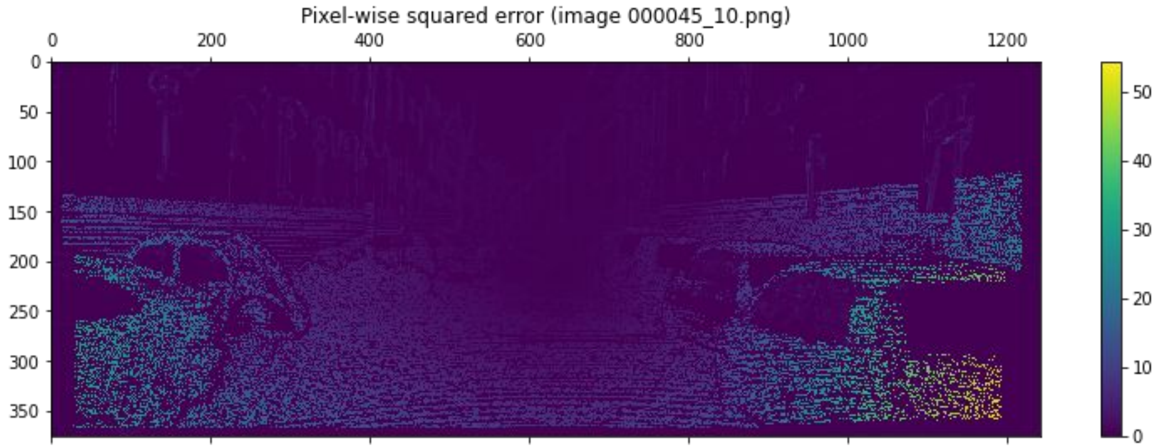


T3.3 Analysis (Team 3) - [3/3]

- There are less pixels with more error in image 45 as compared to 157.
- We can see that the motion estimation is much better on the image 157 as the histogram shows.
- In the image 45 we can see more closer and uniform object. This produce a higher error as we are not able to estimate the motion when objects are close due to high disparity.
- In the image 157, there is a huge error in the tree region on the right side of the image. This is due to high texture of the tree.
- On the left side of the image, the error is less. This is because there are no movement objects and the scene is more uniform.

T3.3 Analysis (Team 4) - [1/3]

- Represent the squared error at every pixel



By looking at the ground truth optical flow, the camera seems to be zooming or moving in the z (depth) axis, which causes the pixels far away from the centre to have bigger displacements between frames (higher optical flow).

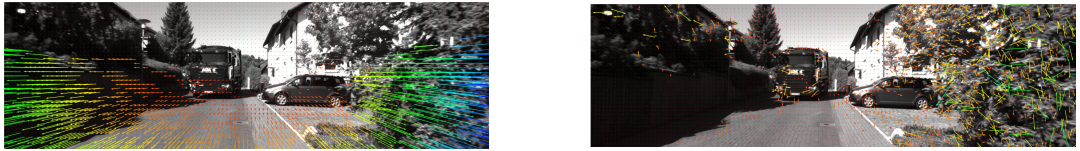
Because of this, we can see in the pixel-wise error plot that the error is the highest on those pixels.

Ground truth and LK predicted optical flow of image 000045.png for reference

T3.3 Analysis (Team 4) - [2/3]

- Represent the squared error at every pixel

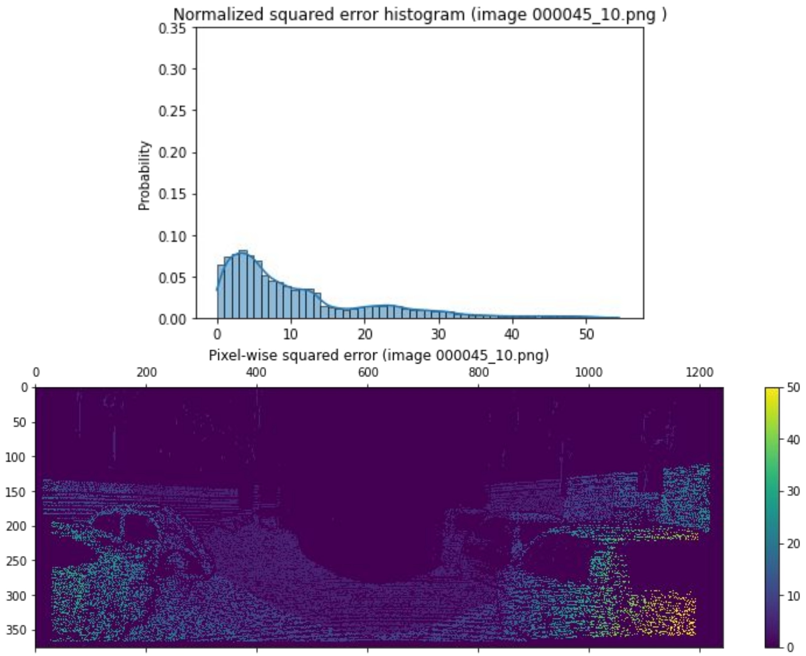
Likewise, the same effect happens in this image as the camera motion is similar. Since the LK algorithm failed to predict most of the values due to the homogeneity of the image where the optical flow is the highest (bold shadows, flat pavement), the error representation is almost the same as the ground truth optical flow.

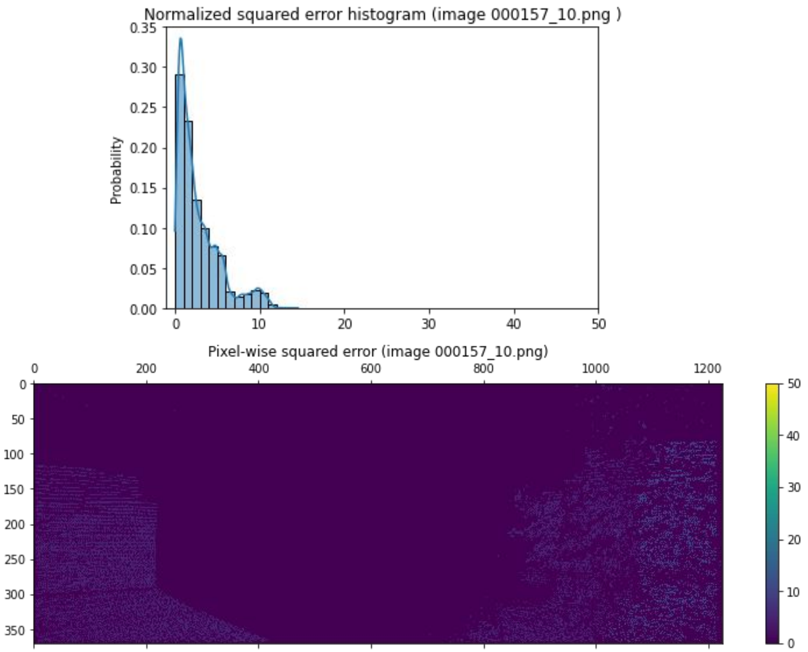


Ground truth and LK predicted optical flow of image 000157.png for reference

T3.3 Analysis (Team 4) - [3/3]

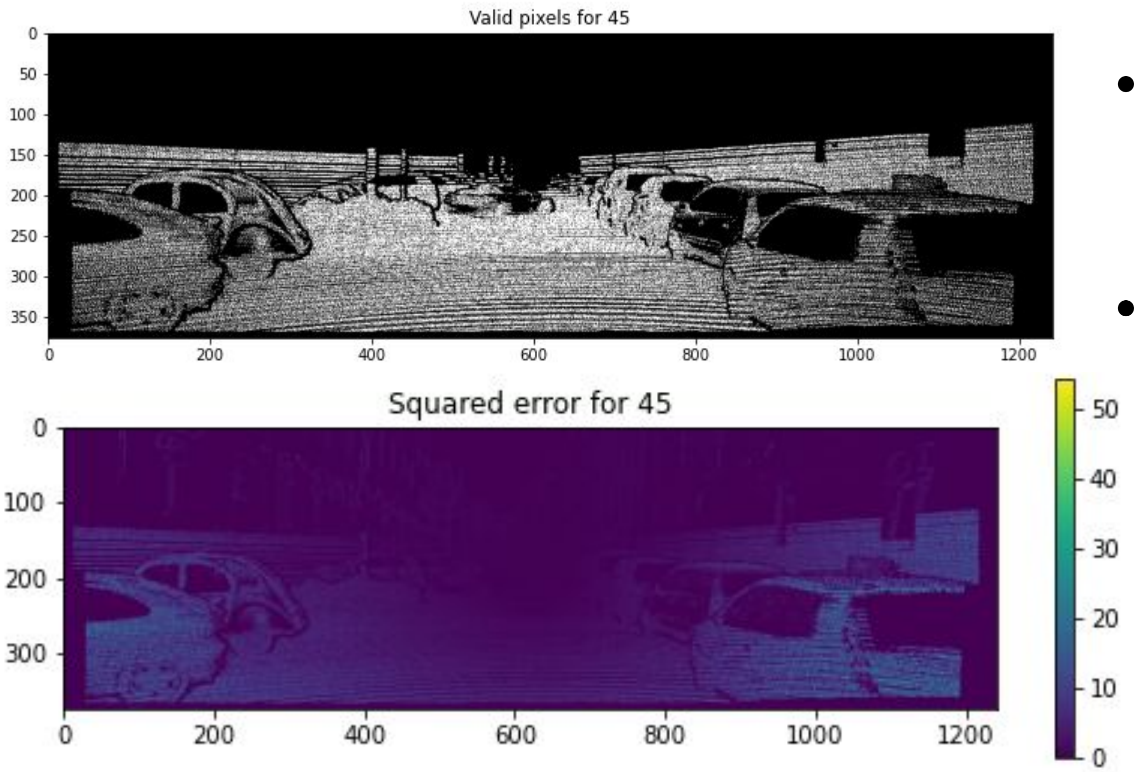
- If we represent the histogram of the error*, we can see that both images have similar distributions, but image 45 has higher error values, which is easier to see in the error plots if we set the same scale to represent colour values. By quantizing the error plot it we can see that the second peak on image 157 (around 10 square error) might belong to the mis-predicted leaves on the foreground.





*Histograms exclude invalid values on the GT

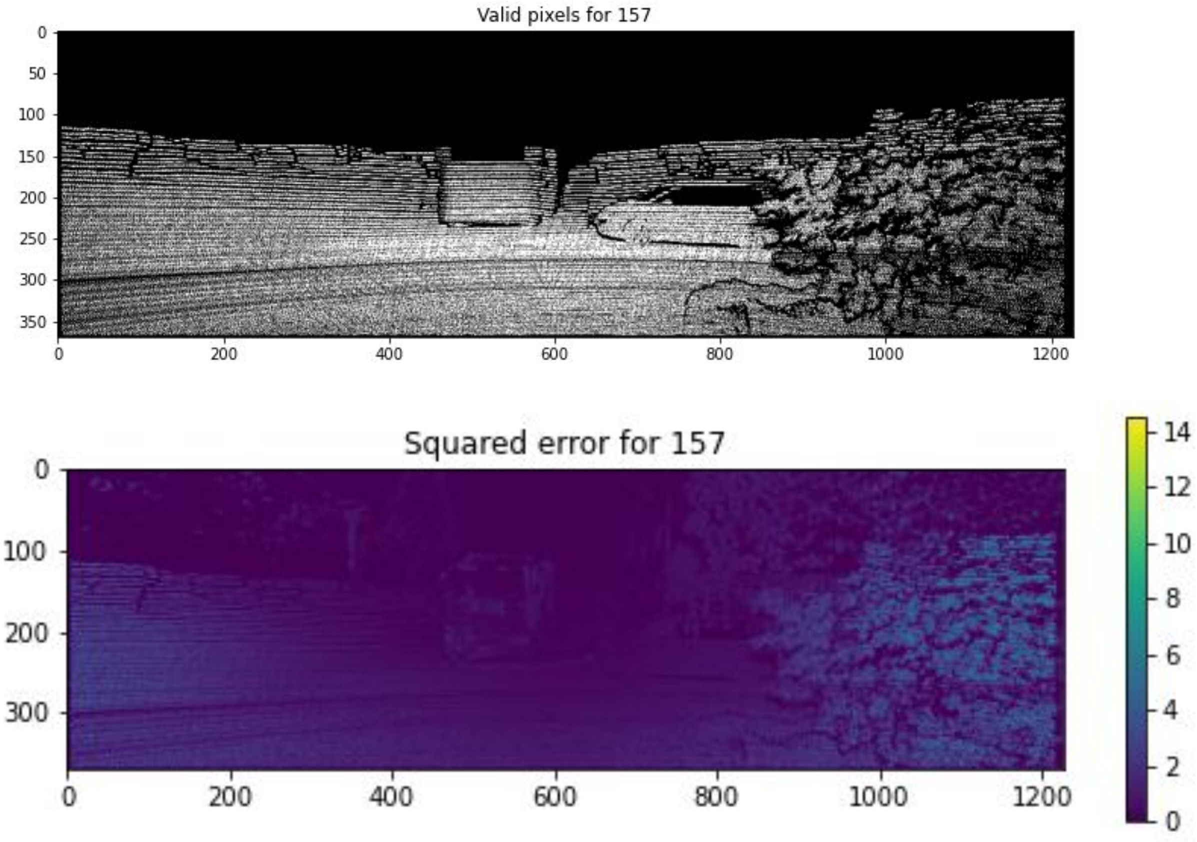
T3.3 Analysis (Team 5) - [1/3]



- We have observed that there are more textureless areas in the road which is quite normal as the color, texture and edges will not change frequently.

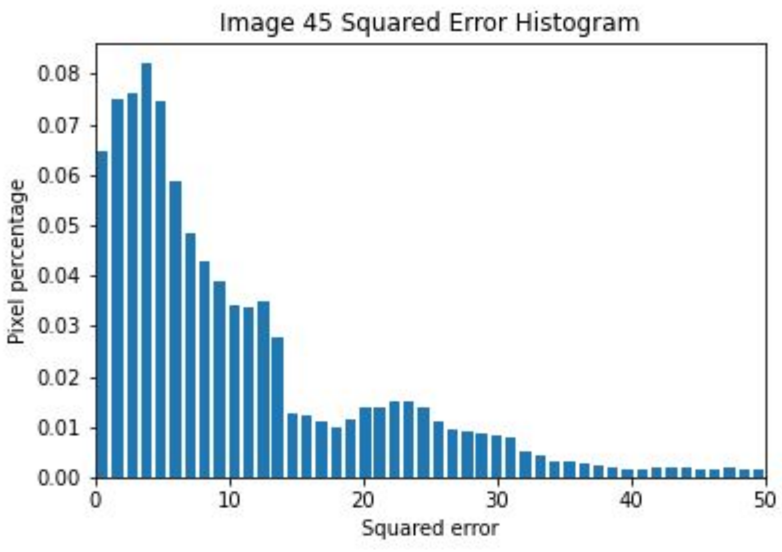
- The MSE is higher on surfaces whose comparing seeing course from the camera community is at a high point w.r.t. the direction of travel. As a result, the MSE is higher on the roadside building facade (bottom) and cars (top). This could be due to the camera's forward motion.

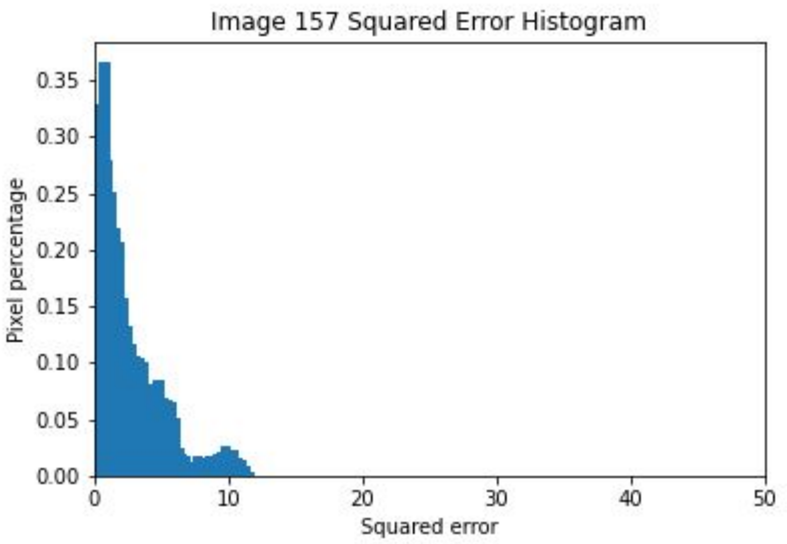
T3.3 Analysis (Team 5) - [2/3]



- On the right side of the image, there is a significant error in the tree region. This is because the tree has a high texture and the objects are closer to the camera (small camera movement, big motion).

T3.3 Analysis (Team 5) - [3/3]



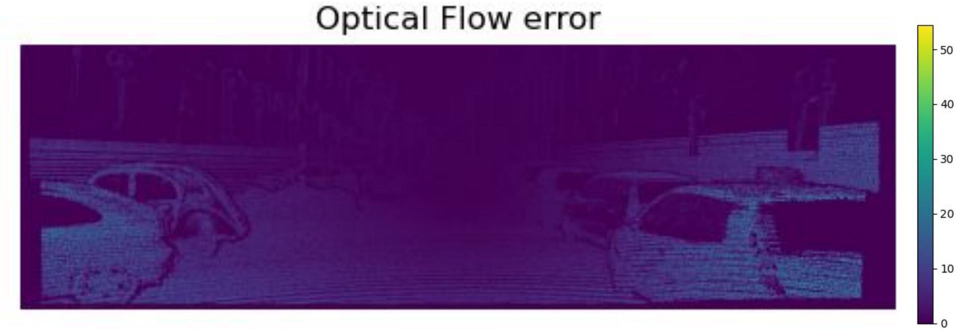


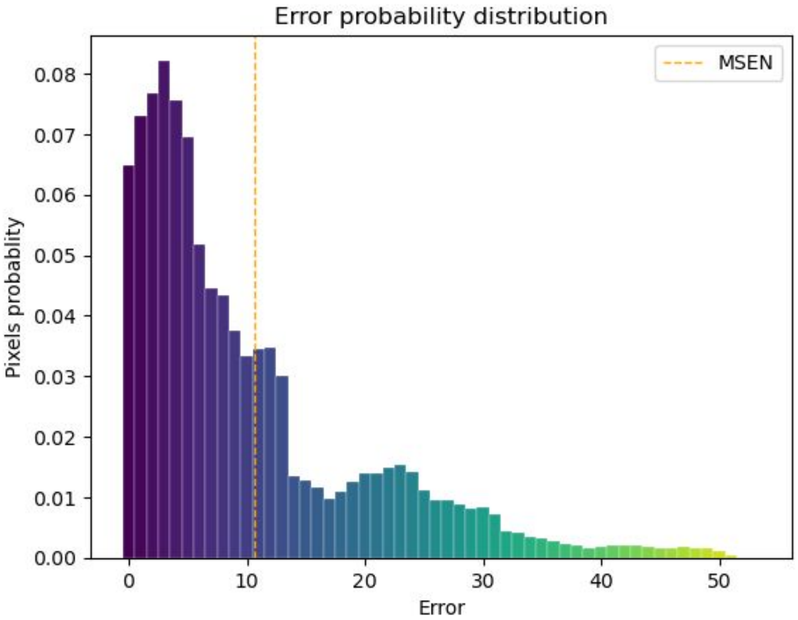
- We can estimate the optical flow estimation is quite better from frame 157, than frame 45 due sequences with rigid motion where the ground-truth flow is determined by tracking hidden fluorescent texture (See ref).

T3.3 Analysis (Team 6) - [1/2]

Sequence 45

MSEN	PEPN
10.63	78,56%





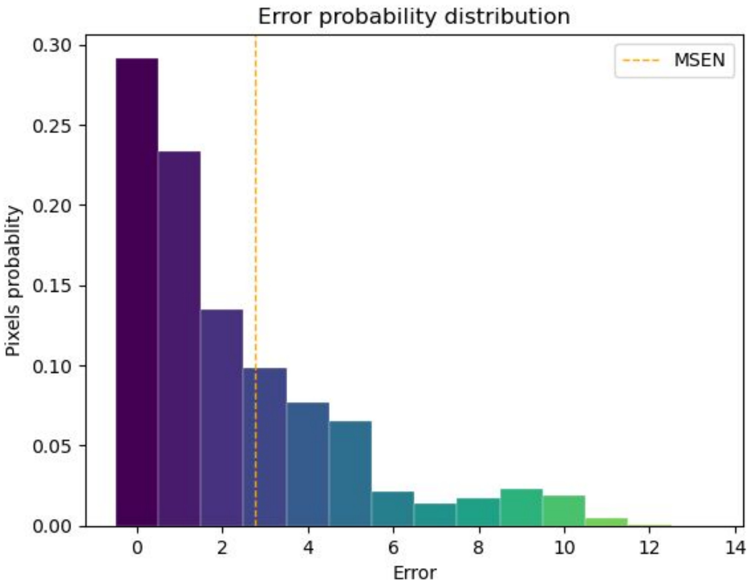
- The error that we are computing is the square error between the GT and the predicted Optical flow in non-occluded areas.
- The diagram shows with darker color the pixels with less error and with brighter colors the ones with higher error.
- In this colormap we see that the error is much large in the zones close to the camera.

- We maintained the colorbar indicating the error in the previous graph and plotted the probability density function of the error.
- Note that the majority of the pixels have low error.
- PEPN is quite high, since we have a lot of pixels with error higher than the threshold = 3.

T3.3 Analysis (Team 6) - [2/2]

Sequence 157

MSEN	PEPN
2.75	34.048%



- Similarly as in the previous sequence, in this colormap we see that the error is much large in the zones close to the camera.
- Note that we have much less error than in the previous sequence, since now the maximum error is 14 instead of 50, and the MSEN has dropped from 10.63 to 2.75.

- Like in the previous sequence, the majority of the pixels have low error.
- In the histogram we can also see that this sequence has much less error between the GT and the predicted Optical Flow than the previous sequence.
- Note that the PEPN error is much lower than in the previous case, since a lot of the error is discarded because it is under the threshold = 3.

T3.3 Instructors' feedback

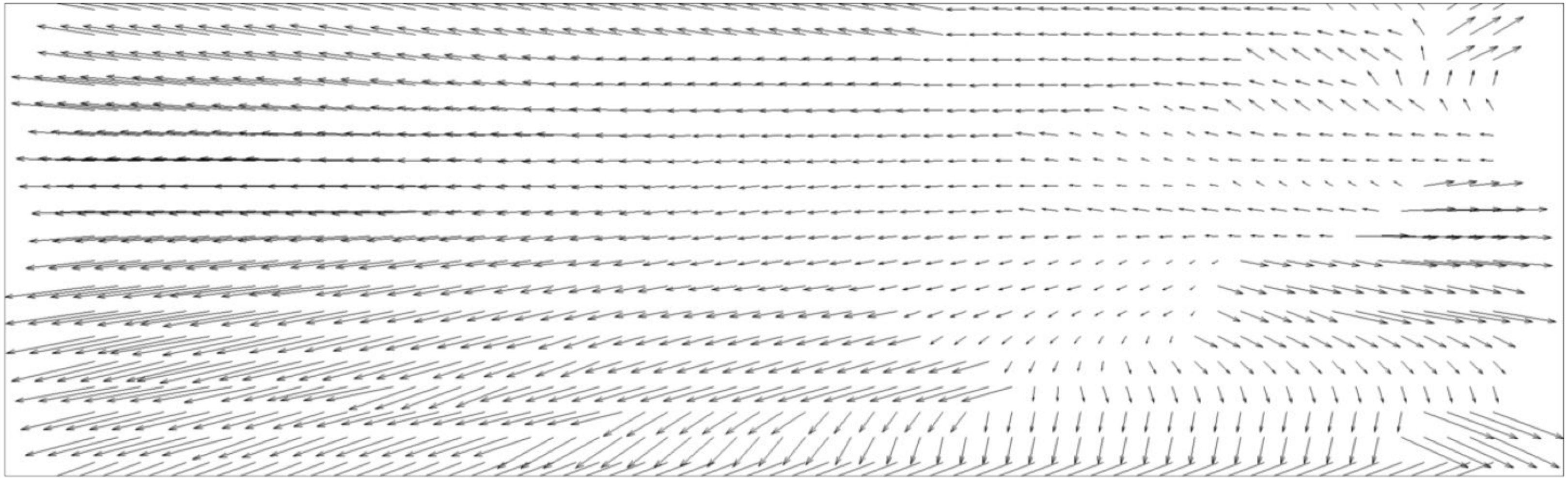
	feedback
Team 1	Use a colour legend to define error dynamic margin. Error coloured map and histogram. Number of bins, maybe too few (no tails) Conclusion → more error in pixels closer to the camera
Team 2	Error and prediction coloured map, also histogram Conclusion → better prediction on textured areas. Higher error on zones at high angle with respect to the camera? What do you mean? Is it closer/further or actually the angle is also important? Good discussion on the “tails” in the histogram.
Team 3	Missing coloured legend to interpret the values. Nice touch to plot the MSEN in the histogram. Comparison between the two “sequences”, good! Conclusion → more error in pixels closer to the camera.

T3.3 Instructors' feedback

	feedback
Team 4	Squared error represented in false color. Coloured legend provided. Not sure about the conclusions. zooming or moving in the z (depth) axis?? Histograms of different frames have the same axis. Good!
Team 5	Squared error represented in false color. Coloured legend provided. Figure showing the valid pixels for GT. Good! Conclusion → "The MSE is higher on surfaces whose comparing seeing course from the camera community is at a high point w.r.t. the direction of travel". Could be clearer … When comparing histograms, it is useful to use the same axes in both cases.
Team 6	Squared error represented in false color. Coloured legend provided. Conclusion→ Max error in regions closer to the camera. When comparing histograms, it is useful to use the same axes in both cases.

T4 Optical flow plot

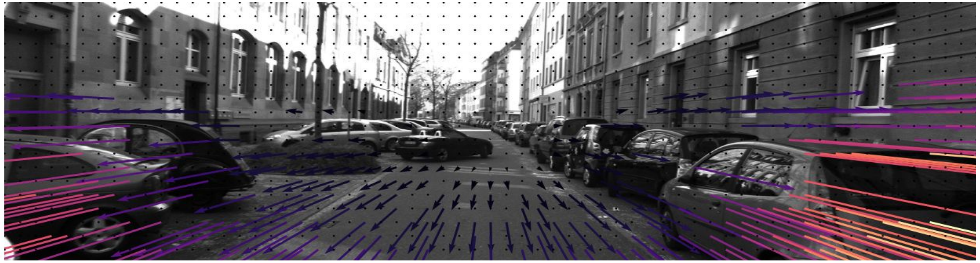
- Plot the optical flow
  - Dense representation -> too many motion vectors
  - Arrows might be confusing, not related to pixels
- Propose a simplification method for a clean visualization.



T4 Optical flow plot (Team X) - Copy & paste (max 3. pages)

T4 Optical flow plot (Team 1) - [1/3]

Using Pyplot's **Quiver:** The magnitude is the length of the arrow and the direction of the arrow encodes the movement direction



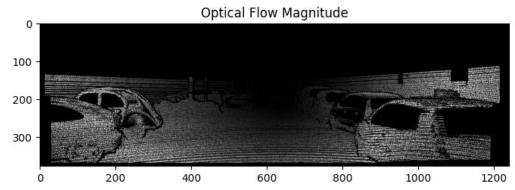
T4 Optical flow plot (Team 1) - [2/3]

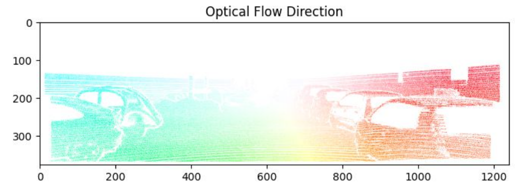
By converting the flow vectors to polar coordinates, we can easily extract the magnitude and the direction

```
# Compute the magnitude and direction of the flow
magnitude, angle = cv2.cartToPolar(flow[..., 0], flow[..., 1], angleInDegrees=False)
```

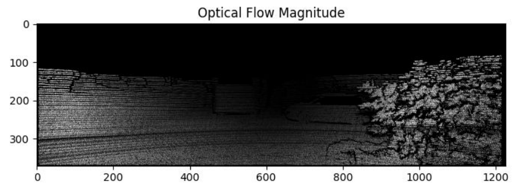
Visualization of the Magnitude and Direction of the OF Ground truth

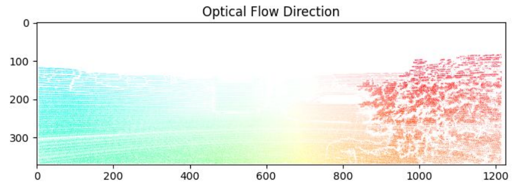
Frame 47





Frame 157

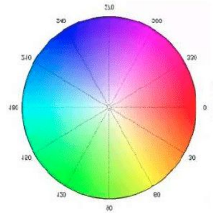




Range of colors:

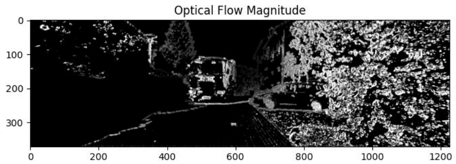
Magnitude:
0 (min) - 255 (max value)

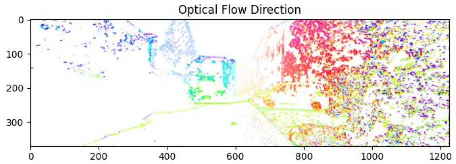
Direction:
HSV Degree as follows

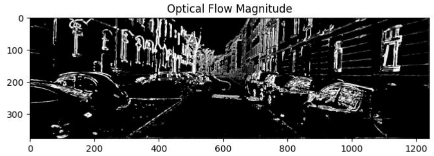


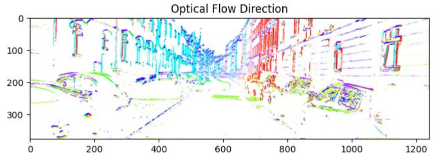
T4 Optical flow plot (Team 1) - [3/3]

We can also see the visualization of the provided predictions. From this we can extract that the predictions are not good enough to properly capture the optical flow of the scene.





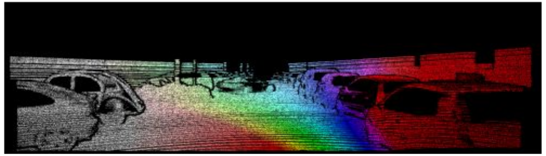


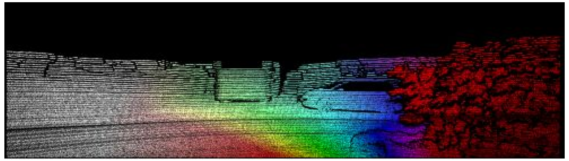


T4 Optical flow plot (Team 2) - [1/3]

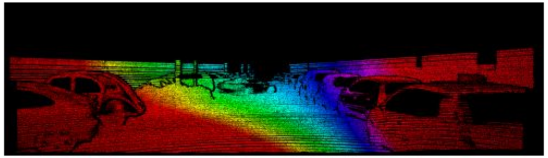
HSV Approach: 2D

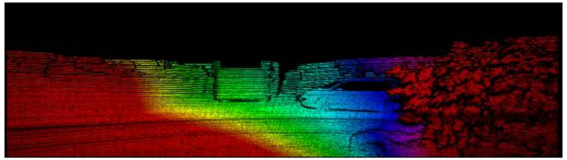
Phase + Magnitude



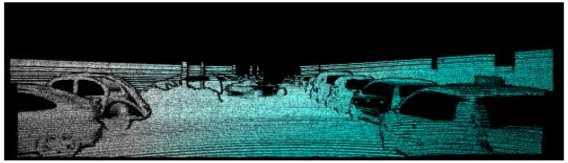


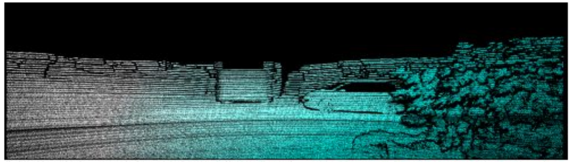
Phase



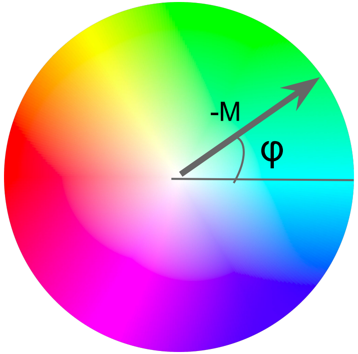


Magnitude





Consider each pixel as a 2D polar coordinate under the HSV space:



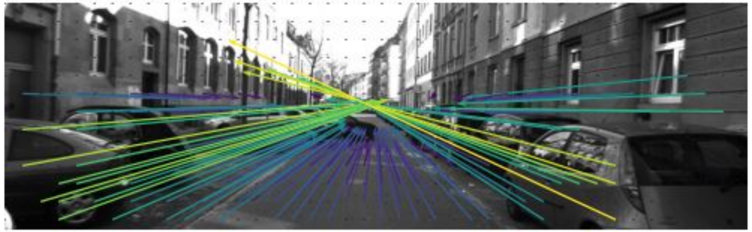
This method can only consider 3D data ($\vec{i}$, $\vec{j}$, *rate-of-change*) in a 2D system (φ, r). We'll move towards 3D representation…

T4 Optical flow plot (Team 2) - [2/3]

Mandatory Vector Fields: 2.5D

Pixel motion:

Camera motion:



This method can now project 3D representations in a 2D plane. We can both visualize the motion field with the projected plane as reference (top) resulting in the standard approach with the pixel motion; nevertheless, this pixel motion is caused by a camera motion* (bottom), we can also set as the reference the projective plane (approximated as the camera) in order to obtain the projective plane motion estimation.

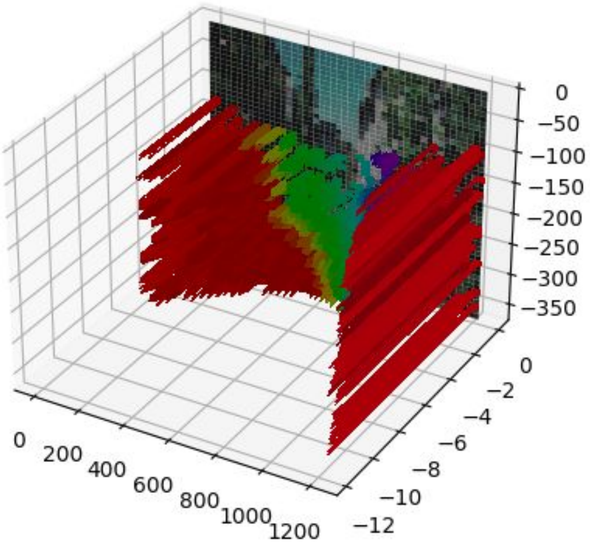
* just in this context

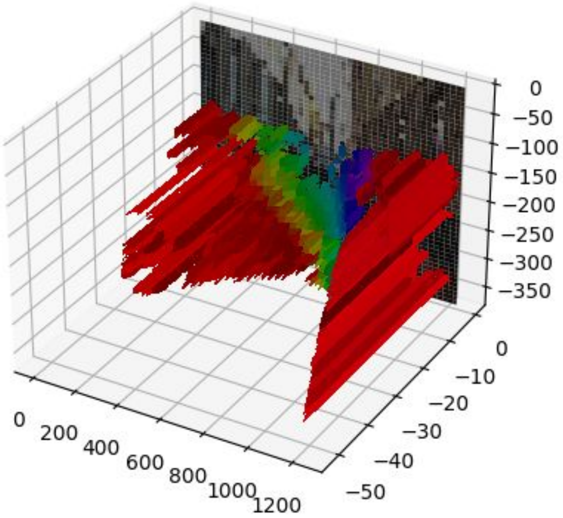
T4 Optical flow plot (Team 2) - [3/3]

HSV + Vector Fields: 3D Depth Reconstruction

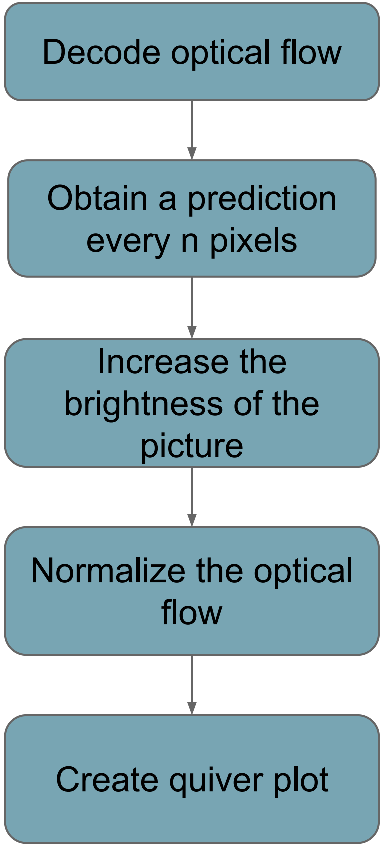
Due to the parallax effect we can correlate 2D motion data (such as pixel displacement calculated by optical flow algorithms) with 3D depth.

Assuming constant velocity, closer objects appear to approach faster under 2D projections. In other words: its pixel displacement is bigger.





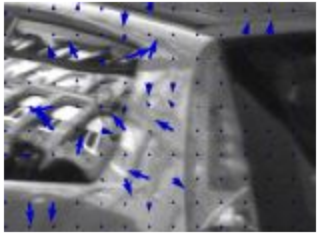
T4 Optical flow plot (Team 3) - [1/3]



- We implemented a function to plot the optical flow from the kitti dataset using quiver to plot the arrows.
- In the dataset every value of the optical flow has 3 values (u,v,mask oclusion). For this task we don't need the last value.
- As the data is too dense, we had to get a prediction every n pixels (10) instead of using it all.
- The magnitude of the movement is the length of the arrow that is also shown in a color representation (bluish is smaller movement, reddish is bigger movement).
- The direction of the arrow points the direction of the movement.

T4 Optical flow plot (Team 3) - [2/3]

Prediction of the optical flow from frame 45

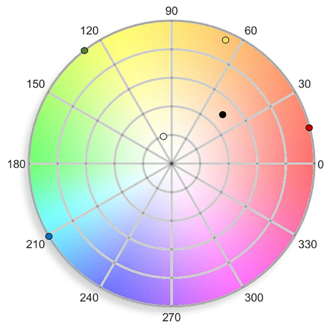


Ground truth of the optical flow from frame 45

We can see that the prediction hasn’t done a good job at computing the optical flow.
Also, we can appreciate that usually in the prediction the magnitude is higher in the edge of the objects.
Due to the parallax effect, the objects that are more near to the camera are the ones that suffer a bigger change.

T4 Optical flow plot (Team 3) - [3/3]

We can obtain a dense optical flow transforming the optical flow to HSV color.
This is done by mapping the polar coordinates to the Hue and the magnitude to the Value.



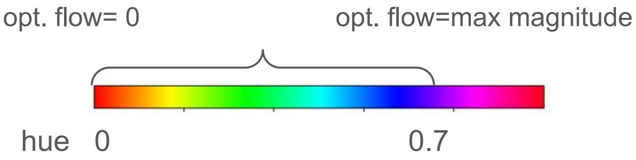
Polar coordinates to Hue

Ground truth of the optical flow from frame 45 using color wheel

T4 Optical flow plot (Team 4) - [1/3]

Quiver plot representation

- Direction represented by the arrow angle
- Magnitude represented by the arrow length and optionally colour*
- We can vary the density of the representation changing the subsampling stepsize

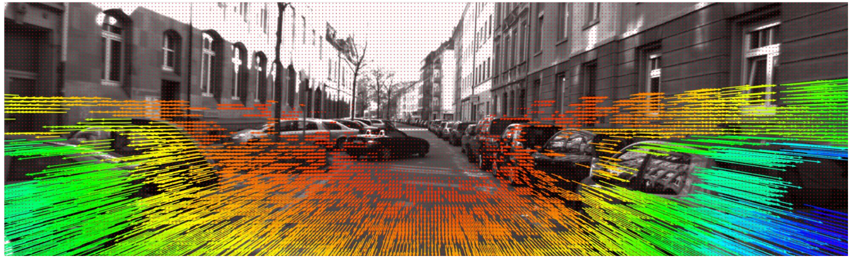


*Colour values are obtained mapping the minimum magnitude to 0 and the maximum to 0.7 of the hue spectrum. The spectrum is clipped to 0.7 so that minimum and maximum values are not too similar in the representation

Quiver plot (stepsize = 10)

Quiver plot (stepsize = 5). Smaller step sizes might allow to observe finer detail, but the representation becomes too cluttered

Quiver plot + hue magnitude (stepsize = 5)

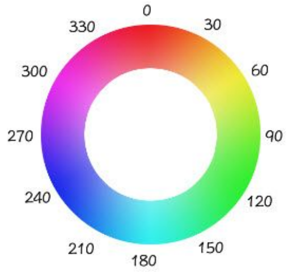


Quiver plot + hue magnitude (stepsize = 10)

T4 Optical flow plot (Team 4) - [2/3]

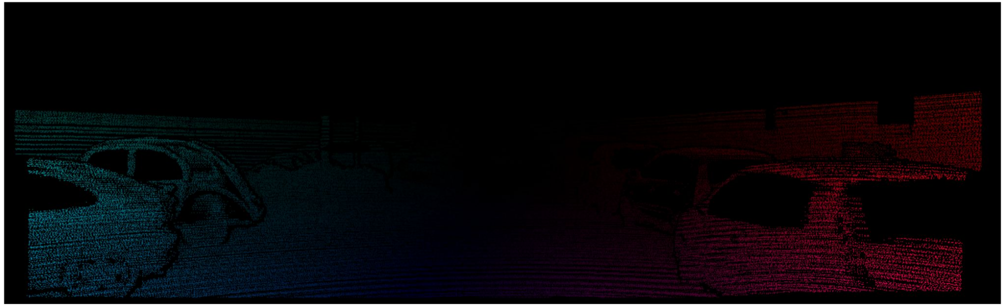
Dense optical flow

- Instead of subsampling the optical flow, we plot all pixels assigning them values in the HSV colour space.
- Angle is assigned mapped hue
- Magnitude is mapped to value



angle

normalized magnitude



Dense optical flow GT '000045_10.png'

Dense optical flow GT '000157_10.png'

T4 Optical flow plot (Team 4) - [3/3]

Optical flow analysis

Quiver optical flow
GT '000157_10.png'

Quiver optical flow
GT '000045_10.png'

Dense optical flow
LK '000157_10.png'

Dense optical flow
LK '000045_10.png'

Quiver optical flow LK '000157_10.png'

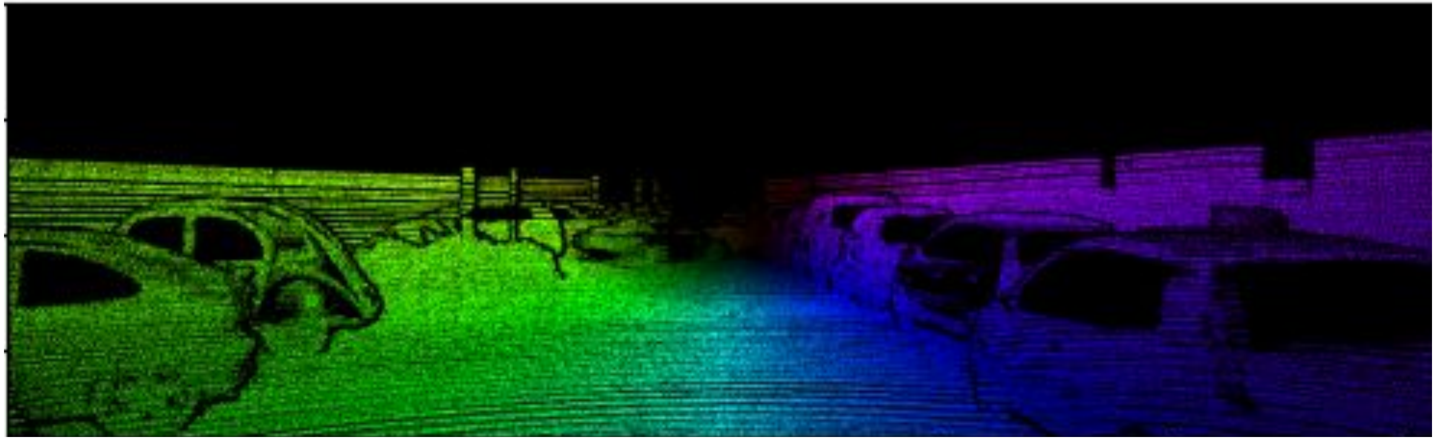
Quiver optical flow LK '00045_10.png'

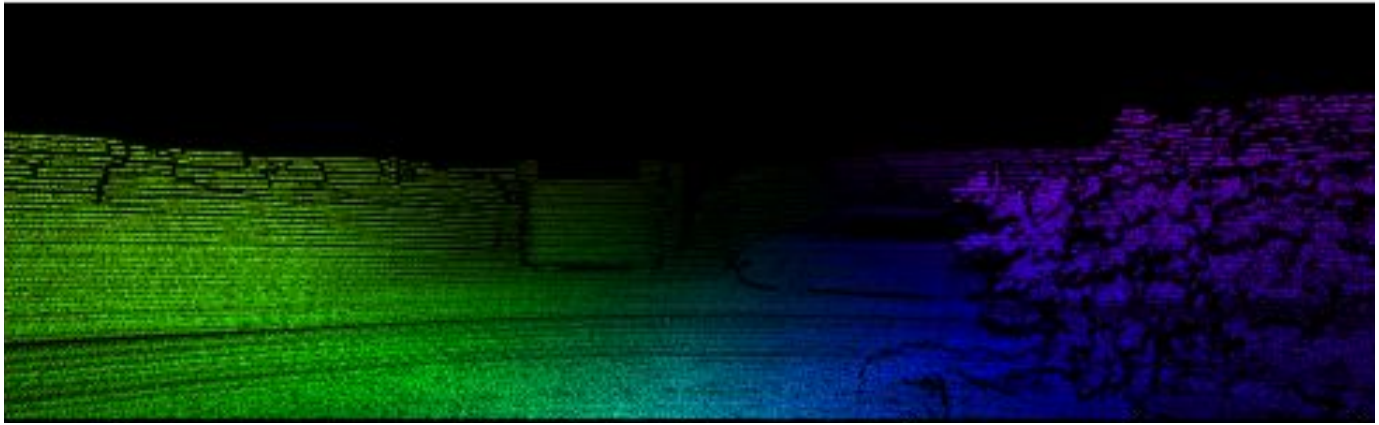
Flat areas such as the pavement were not detected with the LK algorithm, possibly due to their lack of edges/transitions which makes it harder to to estimate the optical flow.
Edges such as the van and fence (image 157) or the windows/shadows (image 45) have a denser amount of non-zero predictions for the opposite reason. Furthermore, repetitive areas such as the leaves (157) were predicted poorly.

T4 Optical flow plot (Team 5) -[1/3] – how do we visualize

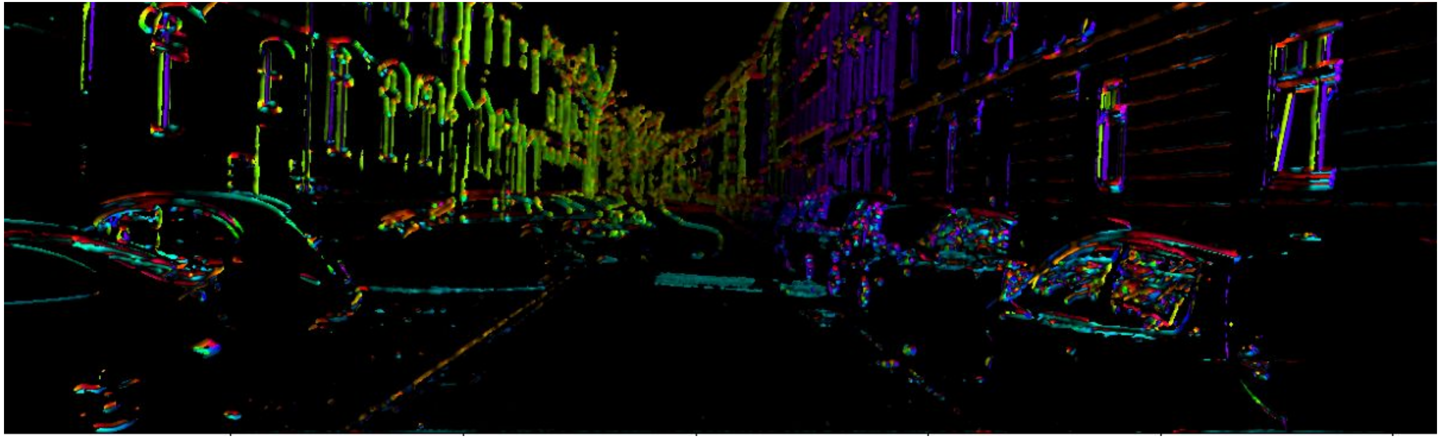
1. Load the OF data (decode a 16-bit .png)
   X-, Y-offsets, validity
2. Compute magnitude using the L2 norm, compute angle using atan2
3. Clip magnitude up to an arbitrary maximum (bound). Good value for this is ~3 (pixels). Smaller bound → more compression, but a brighter image.
4. Create an **HSV** image, where:
   - **H**ue is the angle of an OF vector
   NOTE: be careful with the angle values: OpenCV takes H=[0;180] for uint8 and H=[0;360] for float32. So there is a need to re-map angle value to a relevant range. For example, with angle += 180 (from -180;180 to 0;360)
   - **S**aturation doesn't encode anything. Set it to the maximum for the better visibility.
   - **V**alue (brightness) is the magnitude of an OF vector
5. Convert to an RGB image

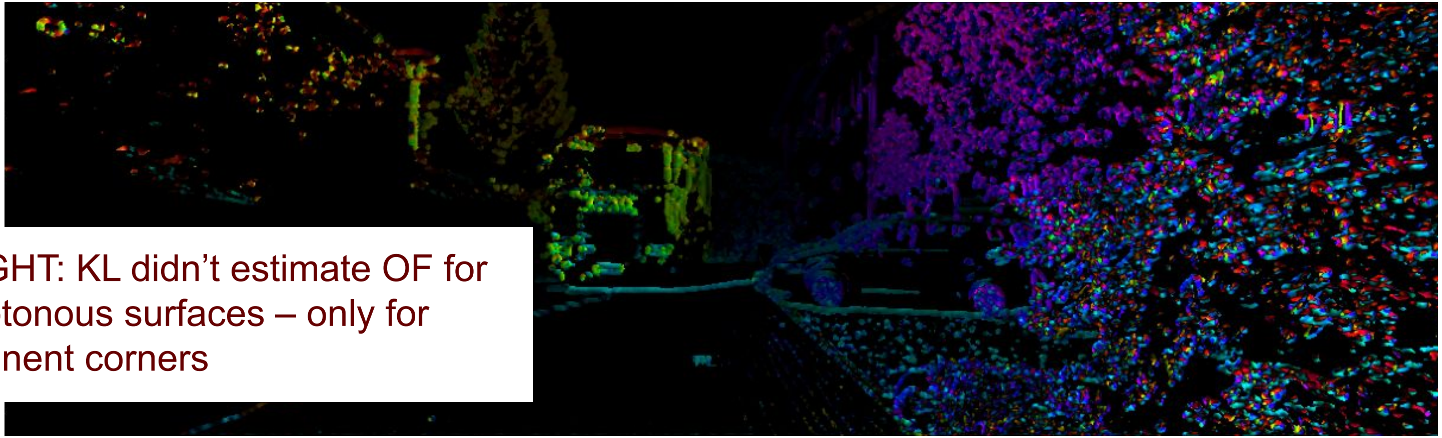
T4 Optical flow plot (Team 5) -[2/3] – Ground truth





T4 Optical flow plot (Team 5) -[3/3] – KL estimation





INSIGHT: KL didn't estimate OF for monotonous surfaces – only for prominent corners

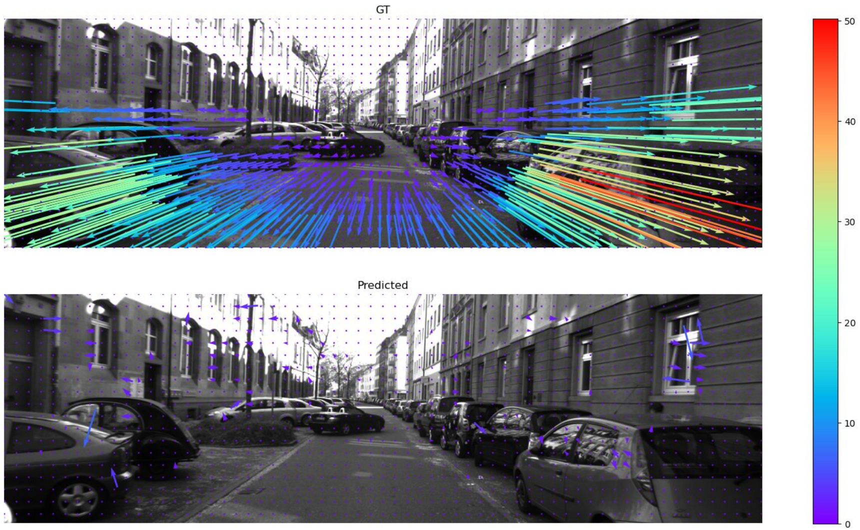
T4 Optical flow plot (Team 6) - [1/3]

As a first approach, we plotted the optical flow with quiver, with step = 15 and scale = 0.1. We maintained the same scale for the GT and the predicted Optical flow to be able to compare it.
In the graph, we see the **magnitude** of the flow represented by the length of the arrow (and also the color, for better visualization) and the **direction** of the arrow encodes the direction of the flow/ movement.
Note that between sequences, the scales change, since the magnitudes are different.

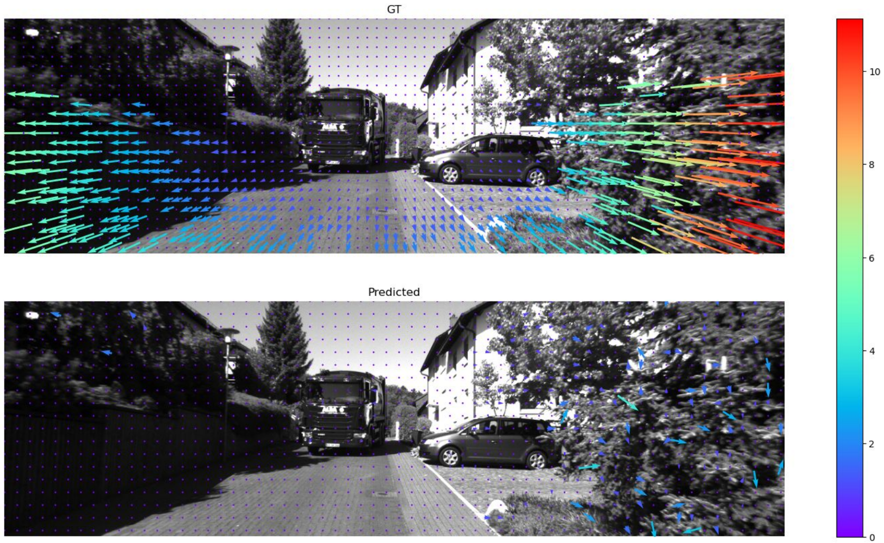
RESULTS

- There is a huge difference between the the GT optical flow and the predicted one.
- As we saw before, in the Sequence 157 flow graph we see that the predictions match better the GT than sequence 45.

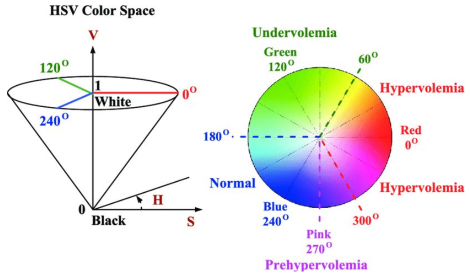
Sequence 45 Optical Flow



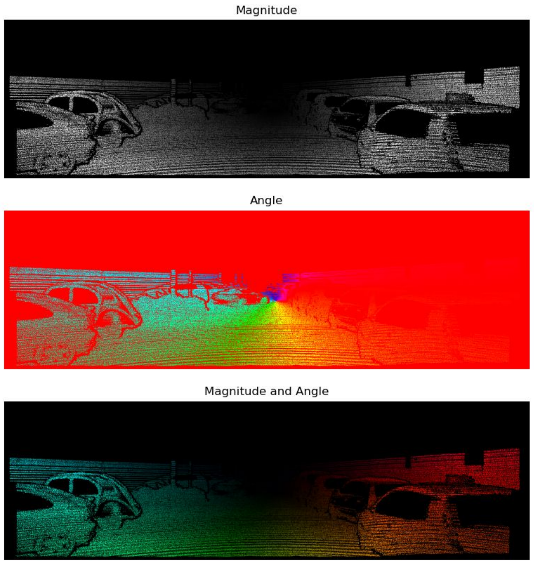
Sequence 157 Optical Flow



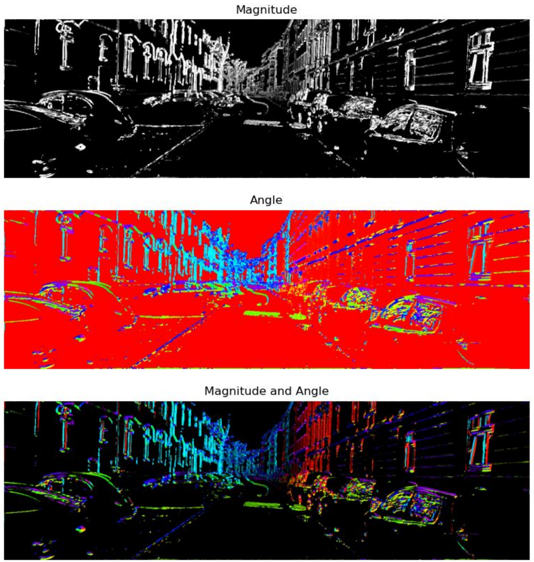
T4 Optical flow plot (Team 6) - [2/3]



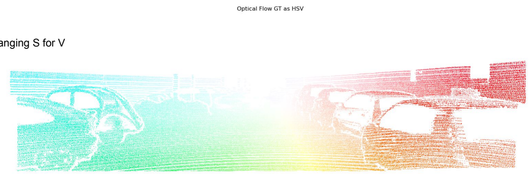
Sequence 45 GT Optical Flow

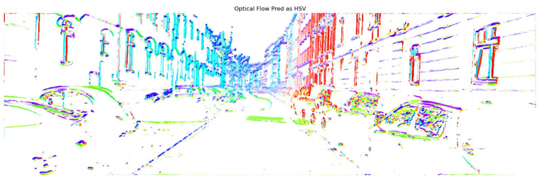


Sequence 45 Pred Optical Flow



Changing S for V





Second approach for better visualization of **dense** Optical Flows working in the HSV space.

First we encoded the data as (based on [1])

- **Direction of the optical** → Hue (color wheel on the right)
- **Magnitude of the optical** → Value (1 [white]: max value, 0 [black]: min value)
- Saturation → set to 255 to see better the colors

However, we also computed a third visualization, just interchanging the V and S channels, so that the plot of the magnitude + direction was more clear (less black, since in this case the magnitude goes from white (0) to more color saturation for the bigger magnitudes.

In both cases we will use only the 0.95 quantile of the magnitude data to help visualization since then outliers or really big magnitudes will not affect in the normalization.

RESULTS

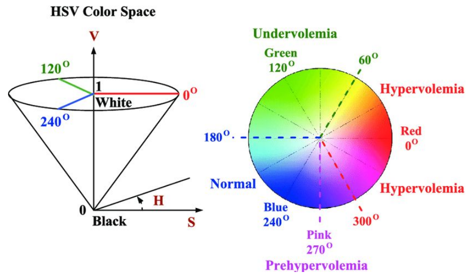
We see that this method enables us to visualize better the Optical flow than when using the arrows.

Here we can also appreciate the difference between the GT and the predicted Optical Flow, so the predictions may not be good enough.

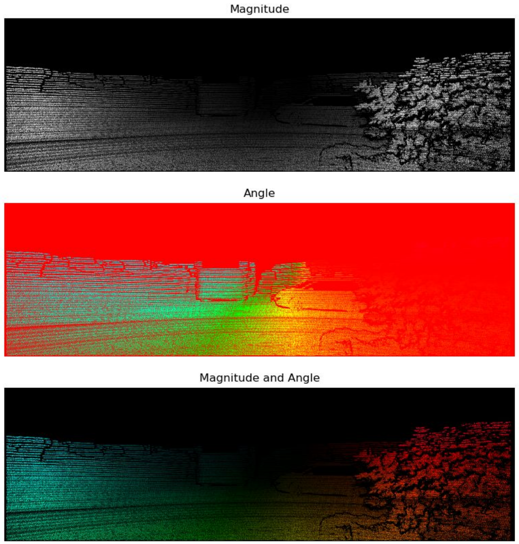
Note that in the predicted Optical flow many areas have a lower magnitude than the GT.

[1] https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html

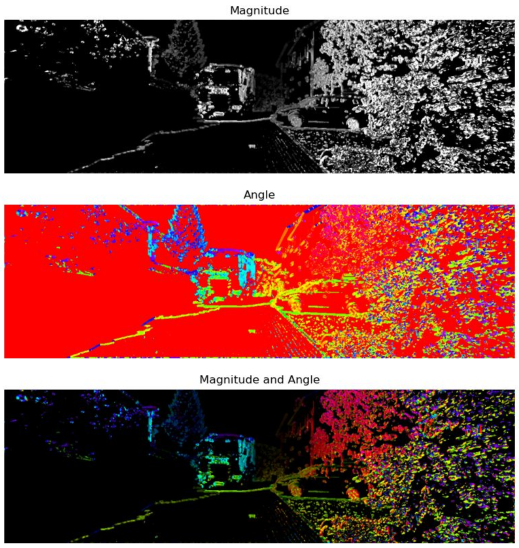
T4 Optical flow plot (Team 6) - [3/3]



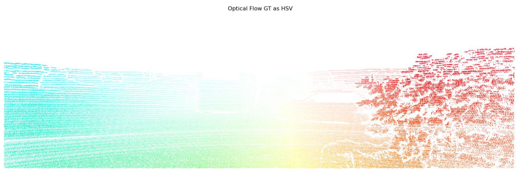
Sequence 157 GT Optical Flow

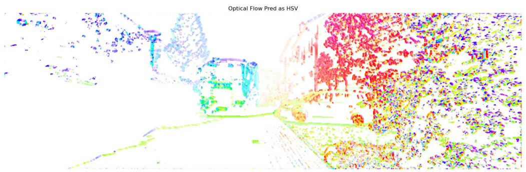


Sequence 157 Pred Optical Flow



Changing S for





RESULTS

Although the GT and the predictions are not really similar, we see that when plotting the magnitude and the angle (last row), in this sequence we have more similar results than sequence 45.

Thus, this matches the previous results, where we saw that for sequence 157 the GT and the predictions were more similar (lower error) than for sequence 45.

It is worth mentioning that using the 95 quantile of the magnitude helped a lot to visualize better the results, since if not, there wasn't much color in the image like we can see here:

T4 Instructors' feedback

	feedback
Team 1	2 representations (quiver + colour map with separate magnitude and direction) Join colour map into a single one.
Team 2	Not clear how the legend relates to the magnitude or phase alone. Need to be better explained. Explain the process to extract vectors in slide 2? Bonus point for the parallax effect and 3D visualization.
Team 3	Nice summary of the steps in slide 1 2 representations (quiver + coloured map) what are the dots in the legend? Do you normalize the magnitude?
Team 4	2 representations, quiver+color for magnitude and color code. Legends provided!
Team 5	One representation, color code. No legend provided.
Team 6	Two representations. quiver+color for magnitude and color code. Legends provided. Different color codes for better visualization. Good