# Supplementary Material for PredStereo: An Accurate Real-Time Stereo Vision System

Diksha Moolchandani, Nivedita Shrivastava, Anshul Kumar, and Smruti R. Sarangi
Indian Institute of Technology Delhi
New Delhi, India
{diksha.moolchandani, anshul, srsarangi}@cse.iitd.ac.in, nivedita.shrivastava@ee.iitd.ac.in

## Abstract

*This supplementary document provides the extra details of our experiments from the main paper and discusses some additional experiments. We first discuss the details of the motivation experiments in Section 1. Subsequently, we discuss the characterization experiments on the GPU and the details of the characterization on the hardware accelerator in Section 2. Next, we discuss the data augmentation details and the configurations of the classifiers in Section 3. We also show the sensitivity analyses of our chosen features. Lastly, we explain the calculation to convert the disparity error into the reduction in the safety buffer time in Section 4.*

## 1. Motivation Experiments

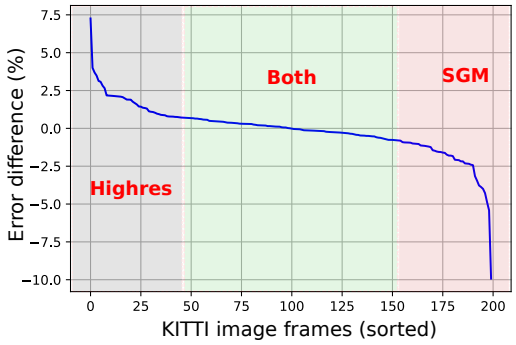


Figure 1. Error difference of SGM and Highres for KITTI 2015 dataset

We motivate our scheme by looking at the results in Figure 1. On the y-axis it shows the difference in the percentage 3-pixel error of SGM and Highres across a set of 200 images from the KITTI-2015 [6] dataset; they are sorted in descending order. We observe that for roughly half the images, the CNN-based algorithm Highres [10] is better than the most accurate traditional algorithm, SGM (Semi-global matching) [2]; the reverse is true for roughly the other half. This debunks a popular belief that CNNs are always the best when it comes to computer vision tasks [4].

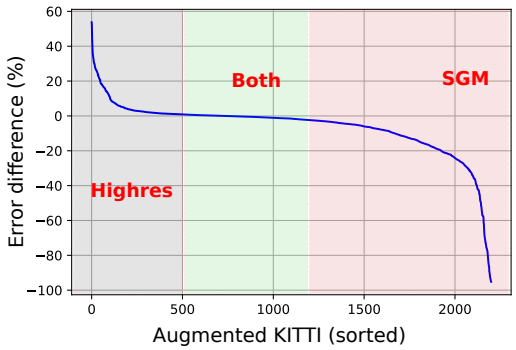


Figure 2. Error difference of SGM and Highres for augmented KITTI dataset

We plot similar data in Figure 2 for the augmented KITTI dataset, where we augment the images in the KITTI dataset with general weather scenarios such as snow, rain, fog, etc. (see Section 3). We observe that for 33% of the images, *Highres* is better than SGM while for the rest of the images, the reverse is true. Moreover, the difference in their errors is much larger (up to 95%) as compared to the difference using the non-augmented dataset (up to 9%). This suggests that CNN-based models are not easily generalizable to changing weather scenarios and consequently their accuracy drops.

Note that in both the experiments, the individual errors of Highres and SGM are not large – for KITTI, the average 3-pixel error is 2.66% for Highres and 2.58% for SGM; for augmented KITTI, it is 13% for Highres and 6.8% for SGM. Thus, the large difference in the error in the plots is due to one algorithm performing poorly on a frame while the other performs fairly well. If we choose the more accurate algorithm for each frame, the average 3-pixel error becomes 5.28% for the augmented case, which is lower than the individual errors of Highres and SGM.

Hence, we argue that any reliable high-performance stereo-vision system needs to have a high-level predictor

| Model | MACs ($10^9$) | Dataflow Accelerator (scaled to 16nm [3]) | | | | | | GPU Tesla P100 | | Error(%) |
|---|---|---|---|---|---|---|---|---|---|---|
| | | RS | | OS | | WS | | | | |
| | | Time (s) | Energy (J) | Time (s) | Energy (J) | Time (s) | Energy (J) | Time (s) | Energy (J) | |
| PSMNet | 980 | 2.10 | 0.70 | 3.22 | 1.46 | 3.22 | 1.15 | 0.47 | 47 | 0.8 |
| GwcNet | 1170 | 2.44 | 0.85 | 3.82 | 1.79 | 3.82 | 1.43 | 0.35 | 43.82 | 0.8 |
| DeepPruner | 775 | 1.74 | 0.72 | 2.68 | 1.15 | 2.68 | 0.92 | 0.052 | 5.72 | 1.1 |
| Highres | 55 | **0.09** | **0.048** | 0.19 | 0.089 | 0.19 | 0.070 | **0.039** | **2.19** | 2.6 |

Table 1. Execution time per frame, energy consumption, and error for CNN-based stereo algorithms

| Model | KITTI-15 test dataset | | KITTI-15 train dataset | |
|---|---|---|---|---|
| | Error (D1-all) | Time (s) | Error (D1-all) | Time (s) |
| MC-CNN | 3.89 | 67 | 9.2 | 11.39 |
| PSMNet | 2.32 | 0.41 | 0.8 | 0.47 |
| DeepPruner-fast | 2.59 | 0.06 | 1.5 | 0.037 |
| DeepPruner-best | 2.15 | 0.18 | 1.1 | 0.052 |
| AnyNet | 6.8 | 0.097 | 6.1 | 0.019 |
| GANet-deep | 1.81 | 1.8 | 7.3 | 4.59 |
| GwcNet | 2.11 | 0.32 | 0.8 | 0.35 |
| Highres | 2.14 | 0.15 | 2.6 | 0.039 |

Table 2. 3-pixel error and execution times for CNN-based stereo algorithms

(selection predictor) that intelligently chooses between algorithms such as SGM and Highres (dynamically based on the ambient environment). Moreover, a confidence predictor in conjunction with the selection predictor is needed to provide a confidence on the disparity estimation accuracy of these algorithms. This confidence measure can be used to trigger higher-level driver actions in case of a low-confidence prediction to avoid human fatalities.

## 2. Characterization of CNN-based Stereo Workloads

Table 2 shows the 3-pixel error and the inference time of the popular CNN-based stereo vision algorithms. The 2nd and the 3rd columns show the average 3-pixel error and execution time per frame that are reported by the original papers on the KITTI-2015 test set. The underlying hardware is different for all the algorithms and hence, the execution time numbers are not directly comparable. Thus, we performed our own experiments to calculate the error of disparity estimation and the execution time (shown in columns 4th and 5th) of the algorithms on the same underlying hardware, Nvidia Tesla P100 GPU. We used the pretrained models available on the official Github repositories and tested the model on the KITTI-2015 [6] training dataset (because of the unavailability of the ground truth for the test dataset and the submissions to the evaluation server allowed for the new algorithms).

We observe in Table 2 that the errors on the training set are lower than the errors on the test set for all the models except MC-CNN and GANet. This is intuitive since we use the KITTI pretrained models for all the algorithms. The significant difference in the error and execution times of GANet in the published and our version is primarily because the published numbers are obtained by running the model on the downsampled images, and performing speed related optimizations (not open source). For the comparison of the execution times, we consider the 5th column of the table because these are obtained on the same underlying hardware, Nvidia Tesla P100. We find that the execution times per frame for MC-CNN and GANet-deep are prohibitively large.

From this experimentation we choose the most accurate algorithms for characterization on the CNN accelerator simulator, *Timeloop*: PSMNet, GwcNet, Highres, and DeepPruner. We discard MC-CNN and AnyNet because they show high errors in the estimation of disparity for both the published results and the results from our experiments. Even though GANet shows low error in the published results, we discard it from further characterization because of its high error in our experiments.

Table 1 shows the inference time and the energy consumption (on KITTI images) per frame of the chosen algorithms on the CNN accelerator with different dataflows and the Nvidia Tesla P100 GPU. Upon analyzing the four algorithms, we find that GwcNet has the highest execution time and worst energy efficiency (see Figure 3), while having similar accuracy as PSMNet. Hence, it can be removed from the final algorithms. We thus show the further results and choose the best algorithm from PSMNet, DeepPruner, and Highres. This is shown in the main paper.

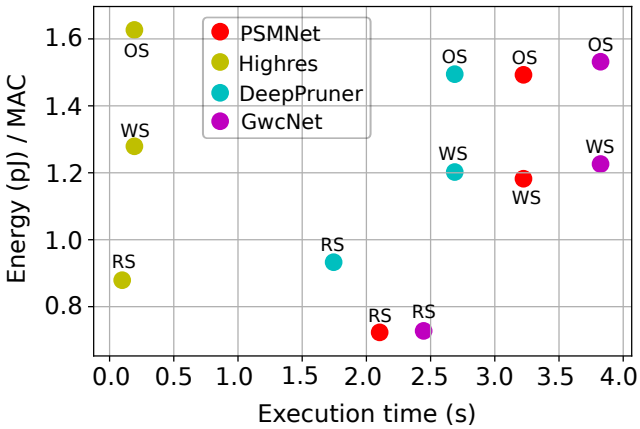


Figure 3. Energy efficiency vs execution time


Figure 4. Example of data augmentation on KITTI dataset. Annotations show the algorithms that achieve better accuracy on the frame

## 3. Design Choices for the Predictor

Table 3 shows the features and their description that we extract from the stereo image pairs to train the predictor.

| Features | Description |
|---|---|
| %Dark regions [9] | Percentage of regions that are dark and provide no useful information |
| Perceived brightness [1] | Quantifies the contrast, color, and reflectance of an image in terms of the weighted sum of the ⟨R, G, B⟩ channels |
| Contrast [7] | Quantifies the local variations of the intensity in an image. |
| Homogeneity [7] | Measures the closeness of equivalent gray-scale levels in an image |
| SSIM (structural similarity index) [8] | Quantifies the similarity of two images in terms of luminance, contrast, and structure. |

Table 3. Features and their description

### 3.1. Data Augmentation

In order to make our predictor generalize to different weather scenarios, we created a new version of the KITTI dataset by augmenting it with images capturing several weather conditions. Figure 4[1] shows an example of the different data augmentations used in our work. We also annotated the figure with the different algorithms that performed well in terms of estimating the disparity on different images.

[1]read this on a color display or take a color printout

### 3.2. Configurations of the Competing Classifiers

Table 4 shows the best configurations for different popular classifiers that were compared to find the best classifier for the selection predictor. We optimized the configurations of these classifiers for accuracy.

| Predictors | Scikit configuration (best) |
|---|---|
| MLP | hidden layer=4, neurons=150, 100, 150, 100, activation=relu, learning_rate=adaptive |
| SVM | gamma=scale, kernel=rbf |
| KNearestNeighbor | n_neighbors=10, algorithm=kd_tree |
| LogisticRegression | solver=liblinear |
| Adaboost | n_estimators=50, learning_rate=0.8 |
| DecisionTree | max_depth=5, min_samples_split=3 |

Table 4. Configurations of the predictors

We performed exhaustive experiments with these predictors and different feature combinations. Based on the experiments, we plotted Figure 7 that shows the accuracy of the different predictors for two types of feature sets: perceived brightness + dark regions (PBDR), and all features. We observed that a decision tree with PBDR achieves the

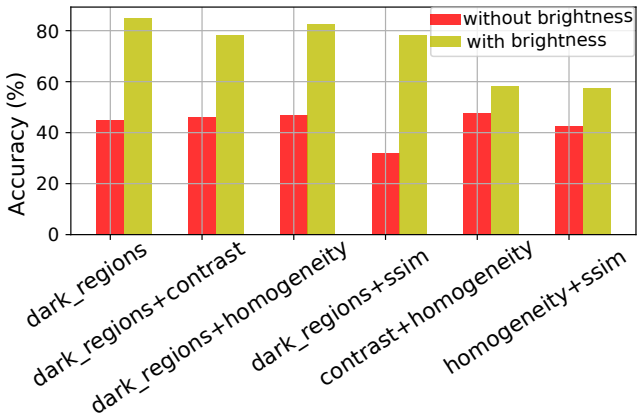


Figure 5. Sensitivity of the predictor to brightness

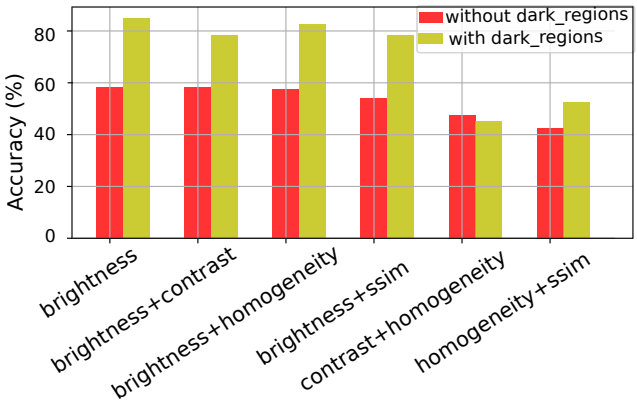


Figure 6. Sensitivity of the predictor to the percentage of dark regions

best accuracy. We also tried designing CNNs for predicting the accuracy but discarded the idea considering the time and energy overheads.

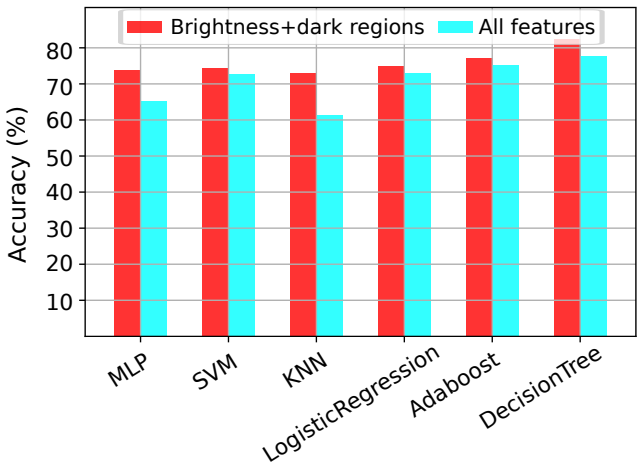


Figure 7. Accuracy of different classifiers

### 3.3. Sensitivity Analysis of the Predictor

Figures 5 and 6 show the sensitivity of the selection predictor with respect to the feature combinations. We observe that with the addition of brightness to a combination of features, the accuracy improves by $11 - 40\%$. Similarly, with the addition of the percentage of dark regions to the feature combinations, the accuracy improves by $10 - 28\%$, with the exception of the feature combination of contrast and homogeneity. This shows that the chosen features are able to accurately capture the texture and occlusion information of the frames.

### 3.4. The Confidence Predictor

As explained in the main paper, we formulate the confidence prediction as a regression task, where the confidence is a value between 0 and $100\%$: defined as $100 - \%3 - pixelerror$. Table 5 shows the mean absolute error (MAE) of the predicted confidence value by the decision tree based confidence predictor for different weather scenarios. The MAE varies between $0.25\%$-$7.3\%$, suggesting that the confidence is predicted accurately and can provide deterministic cues to switch to higher level control.

| | Cloud | Rain | Fog | Snow | Frost | Dropout | Kitti |
|---|---|---|---|---|---|---|---|
| MAE | 3.7 | 2.9 | 5 | 3.6 | 7.3 | 0.36 | 0.25 |

Table 5. Mean Absolute Error of the confidence predictor for different scenarios

## 4. Evaluation Details

### 4.1. Analysis of the Disparity Estimation Error in a Self-Driving Scenario

In this section, we analyze the effect of error in the disparity estimation in a self-driving scenario. Any car has to maintain a half speedometer distance (equivalent to 1.8 seconds) [5] from the vehicles in front of it to allow a safety buffer time. Considering the average speed of a car to be 70 km/hr, the safe braking distance that a vehicle should maintain is 38.8 m. Now let us calculate the error in estimating the depth using the error in disparity. As explained in Section 2 of the main paper, depth is inversely proportional to disparity. Differentiating the equation from Section 2 with respect to disparity ($d$), we get $\partial Z = -\frac{(b \times f).\partial d}{d^2}$. Replacing $d$ by $f \times b/Z$, we get $\partial Z = -\frac{Z^2.\partial d}{b \times f}$. This equation relates the error in depth with the error in disparity. For the images in the KITTI dataset, $f = 721$ pixels and $b = 0.54$ m.

As an example, we consider an image (KITTI training image 29) with the snow augmentation. We consider the car in the scene in image 29. The real depth of the car is 46.29 m from the camera. We obtain a disparity error of 1.74 pixels using *Highres* and 0.37 pixels using SGM for the car pixels. This translates to 9.53 m and 2.02 m error in depth using *Highres* and SGM, respectively. Now using the

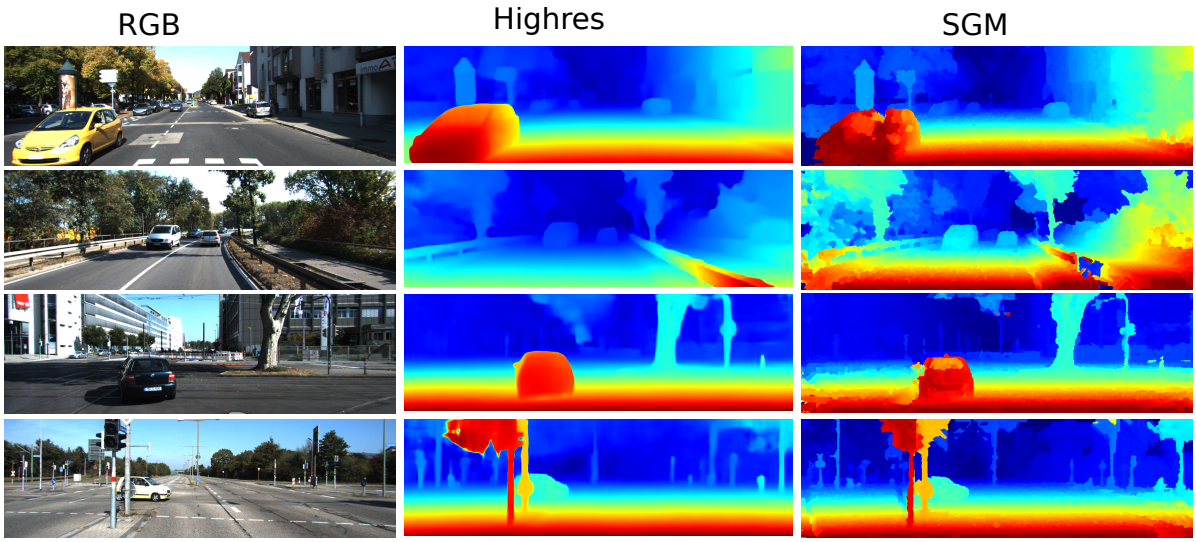


Figure 8. Results of disparity estimation on KITTI-2015 training images. The first column shows the left image of the stereo image pair. The second and the third columns show the disparity map obtained by Highres [10] and SGM [2], respectively.

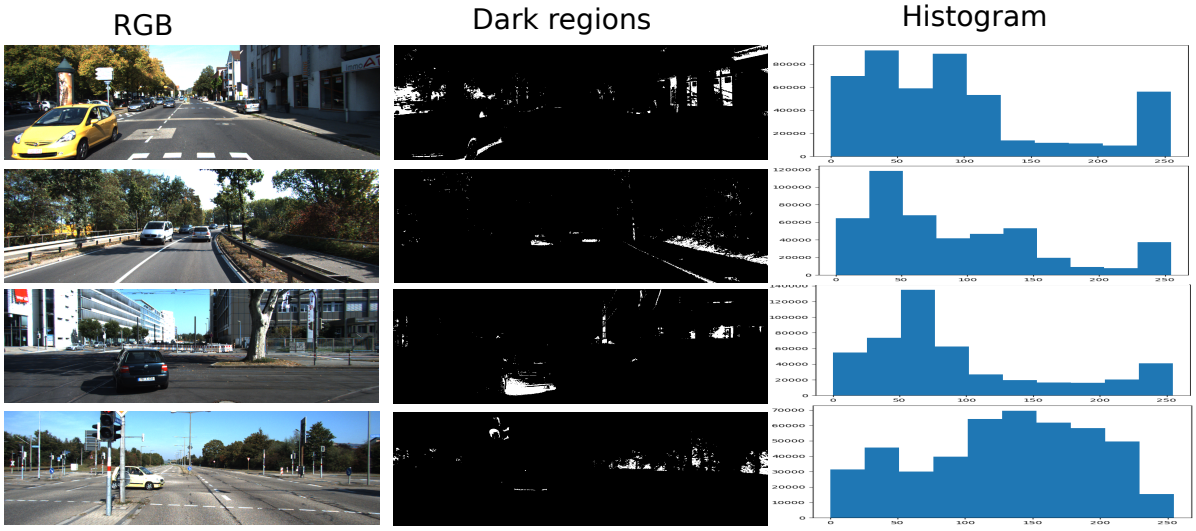


Figure 9. Visualization of some features of KITTI-2015 training images. The first column shows the left image of the stereo image pair. The second and the third columns show the dark regions (in white) and the histogram of the grayscale image, respectively.

*Highres* scheme, we will consider that the vehicle in front is at a depth of 55.82 m and we assume we have a margin of 17.02 m (considering the braking distance of 38.8 m) and can safely accelerate. However, in reality after covering this margin, the real depth would have become 29.27 m (46.29 m - 17.02 m), which reduces our safety buffer time to apply brakes to 1.3 s. Contrarily, using *PredStereo* to predict that SGM is better for the current image frame, we would have reduced the safety buffer time to 1.7 s. A similar situation is observed for the snow-augmented image 182. There the depth error is as large as 44 m using *Highres* while it is only 0.79 m using SGM.

### 4.2. Qualitative Evaluation

Figure 8 shows the disparity maps of some images taken from the KITTI-2015 training dataset. We show these results on the KITTI-2015 training set because of the unavailability of the ground truth for the test dataset and the submissions to the evaluation server allowed for the new algorithms. We use the same evaluation script as available on the KITTI evaluation server.

Figure 9 shows the features of some images taken from the KITTI-2015 dataset. The first column shows the RGB images. The second column shows the dark regions of the

image in white color. The third column shows the histogram of the grayscale version of the RGB image. It shows the distribution of the pixel intensities in the image. A distribution skewed to the left side indicates the dominance of darker pixels.

### 4.3. Novelty vis-a-vis Related Work

In the last few years, the pendulum of innovation has swung towards the side of CNNs disproportionately. However, off late, there is an increased realization that traditional algorithms have advantages in terms of accuracy particularly when we have complex features and a large amount of occlusion. Hence, we would like to conjecture that hybrid models such as ours will gain prominence in the future.

## References


[1] Sergey Bezryadin, Pavel Bourov, and Dmitry Ilinih. Brightness calculation in digital image processing. In *International symposium on technologies for digital photo fulfillment*, volume 2007, pages 10–15. Society for Imaging Science and Technology, 2007.

[2] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 807–814. IEEE, 2005.

[3] Wei Huang, Karthick Rajamani, Mircea R Stan, and Kevin Skadron. Scaling with design constraints: Predicting the future of big chips. *IEEE Micro*, 31(4):16–29, 2011.

[4] Hamid Laga. A survey on deep learning architectures for image-based depth reconstruction. *arXiv preprint arXiv:1906.06113*, 2019.

[5] Markus Maurer, J Christian Gerdes, Barbara Lenz, and Hermann Winner. *Autonomous driving: technical, legal and social aspects*. Springer Nature, 2016.

[6] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3061–3070, 2015.

[7] P Mohanaiah, P Sathyanarayana, and L GuruKumar. Image texture feature extraction using glcm approach. *International journal of scientific and research publications*, 3(5):1–5, 2013.

[8] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[9] Feng Yang, Luciano Sbaiz, Edoardo Charbon, Sabine Süsstrunk, and Martin Vetterli. On pixel detection threshold in the gigavision camera. In *Digital Photography VI*, volume 7537, page 75370G. International Society for Optics and Photonics, 2010.

[10] Gengshan Yang, Joshua Manela, Michael Happold, and Deva Ramanan. Hierarchical deep stereo matching on high-resolution images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2019.