Elevation Inpainting with MAT: Mask-Aware Transformer*

1st Vasudev Purohit
Automotive Engineering
Clemson University
vpurohi@clemson.edu

1st Benjamin Johnson
Automotive Engineering
Clemson University
bij@clemson.edu

***Abstract*—In this study, we investigate the task of inpainting missing elevation data in 2.5D maps for off-road navigation using a Mask-Aware Transformer (MAT). Missing data in elevation maps, caused by environmental occlusions and sensor limitations, can significantly impact navigation safety and efficiency. We propose a method to generate realistic inpaintings while quantifying the uncertainty of these predictions. Synthetic datasets mimicking occlusion patterns were created using ray-casting techniques, with varying levels of masking intensity. Our experiments demonstrate that correlated forward-facing masks yield superior performance compared to aggressive 360-degree masks, resulting in higher fidelity inpaintings. Additionally, uncertainty quantification using an ensemble of predictions reveals potential safety risks in regions with large occlusions, highlighting areas for future improvement.** *

I. Introduction

In this paper we investigate the task of generating missing elevation information in a 2.5D elevation map for the purpose of off-road navigation. Specifically we train transformer based network /citemat to fill in heavily occluded regions of the map based on contextual information from the scene. The task of off-road navigation is made particularly difficult do to the rugged nature of the terrain which affects: (1) Dynamics of the vehicle which can lead to jittery movements causing blurred images (2) Pose of the vehicle and thus pose of the sensors onboard climbing large rocks could point the camera at the sky giving effectively no usable information (3) Occlusions due to large objects such as trees, boulders, etc.

Beyond the geometric problem of off-road navigation, there also exists a broad range of semantic classes with a varying range of traversabilities. Combined with the lack of structured markings such as lane lines or road signs, the task becomes much more difficult than its on road counterpart. In the traditional local trajectory planning paradigm, a local occupancy map is built using onboard perception. In the case of missing data the space is marked as either free or occupied leading to vastly different results. In the case it is marked as free, the planner tends to plan more aggressive trajectories into unobserved space, leading to potential lethal collisions. On the other hand, marking it as occupied can lead to overly cautious plans with trajectories taking much longer to reach the goal or avoiding occluded space all together. Previous work [1]–[3] has shown that filling in missing information with a data driven approach such as inpainting leads to the planner being able to plan longer, safer trajectories which are much closer to those of a planner with ground truth data available. However, though it has been investigated before, none of these works quantify the confidence of their inpainted maps. We know that neural networks make mistakes, just as a human would, unlike humans though, neural networks don't have an implicit understanding of their own uncertainty leading to potentially lethal decision if the data is taken at face value. [4], [5] argue that considering the uncertainty of the occupancy map at planning time can lead to safer trajectories with fewer replans than the traditional binary approach. Thus in this work we focus on not only inpainting occluded elevation information, but additionally quantifying the uncertainty of the inpainted map, which can then be used by downstream planning tasks.

II. Related Work

Image inpainting has a rich history in computer vision. Inspired by the field of image or art restoration by qualified and unqualified artists alike, its main objective is to repair missing or damaged image data or remove objects and replace with a plausible replacement in its place. In recent years, data driven approaches such as VAEs, GANs, and Diffusion-models have emerged as the state of the art. With the ability to produce life like results on a varied array of data it has become increasingly difficult to distinguish the reals from the fakes. With such a powerful tool at their disposal many works have looked at utilizing image inpainting for the task to filling missing map information for autonomous navigation [1]–[3], [6] .

[1] Uses Pix2PixHD modified with an inpainting-targeted L1 loss. The network is trained on the sparse semantic LiDAR from the KITTI-360 dataset. The point-clouds are projected to the birds eye view and dense 2.5D images are rendered in the generator. They show a significant increase in the length and safety of the paths planned using their inpainted maps as compared with the non inpainted versions. [2] focuses on the task of inpainting digital elevation maps (DEMs) for the purpose of off-road navigation of a legged robot. They take a self supervised approach in training a U-Net style network.

[1]Vasudev Purohit (vpurohi@clemson.edu) is with the Department of Automotive Engineering, Clemson University, Greenville, SC 29607, USA.
[2]Benjamin Johnson (bij@clemson.edu) is with the Department of Automotive Engineering, Clemson University, Greenville, SC 29607, USA.
*Code available at: inpaint-elevation.github.io.

Given an input DEM with occluded regions, a virtual robot is spawned on the map which performs ray-casting over the DEM in a known region. These artificial occlusions are then used to train the network on the actual occluded data using an MSE and total variational (TV) loss. [3] also employs a U-Net style architecture with skip connections, but rather than inpaint only elevation information they inpaint the entire 3D occupancy. Their experiments show that the utilizing the inpainted map lead to smoother and safer trajectories than both assumed free and assumed occupied planing methodologies. All of these works show the benefit of employing data driven inpainting for planning, but fail to quantify the uncertainty of their predictions which could potentially lead to a failed trajectory.

However, there are a plethora of works which utilize uncertainty for planning [7]–[9]. The field of autonomous 3D reconstruction has exploded in recent years thanks largely in part to advancements 3D representation models such as NeRF [10], 3DGS [11], Occupancy Networks [12], and InstantNGP [13]. The general approach in these works to render an array of images from novel view points in the 3D scene. Then based on the uncertainty or entropy in the rendered images, choose the area with highest uncertainty and plan a path to observe that point. This works very well for build rich dense maps of various scenes, however these approaches are very slow and the uncertainty isn't incorporated directly into the planning, but rather as a guide for which areas need to be explored further. Perhaps the closest to our work in terms of the uncertainty quantification is [14]. They to are quantifying uncertainty in the map for the purpose of exploration, but their approach of rendering an ensemble of images of the same location is similar to ours. However, they produce an ensemble with multiple networks, where ours only needs a single network and relies on injecting noise to generate the ensemble.

III. Method

A. Overall Architecture

The overall architecture has been directly borrowed from [15]. The MAT architecture integrates the strengths of transformers and convolutions through a convolutional head, a transformer-based body, a convolutional tail, and a style manipulation module. The convolutional head extracts tokens, which are processed by a transformer body comprising five stages of multi-head contextual attention (MCA) blocks at varying resolutions to model long-range interactions. The resulting tokens are upsampled to the input resolution using a convolution-based reconstruction module. Additionally, a Conv-U-Net refines high-frequency details, leveraging the efficiency and local texture refinement capabilities of CNNs. Finally, the style manipulation module modulates convolutional weights to enable diverse/pluralistic predictions.

B. Data Set Generation

Due to real world data from our testing grounds not yet being available to train with, we synthesized elevation map data. We generated two different types of data with increasing level of complexity/randomness in it. They are both generated using a randomly distributed blob approach.

In the first set, we randomly set the center (X, Y), radius r, and height h for a fixed number of "blobs". Then we apply a Gaussian filter to the generated image. This resulted in the data seen on the left of figure(1). The intent was to represent a field of small trees or boulders, and at first glance it looks as though it could be. However, after training with this data for 24 hours, we were unhappy with the way the model was converging and decided to try synthesizing something a little more natural feeling.

For the second set of data, we used a similar approach. However, rather than use a fixed number of blobs, we used a random number of patches each including a random number of blobs clustered around their centers. The image on the right of figure (1) shows this yields a much more natural feeling data, with patches now representing groups of trees or boulders. In addition to creating a new dataset, we generated correlating masks for each image which we discuss in detail in sectionIII-C.

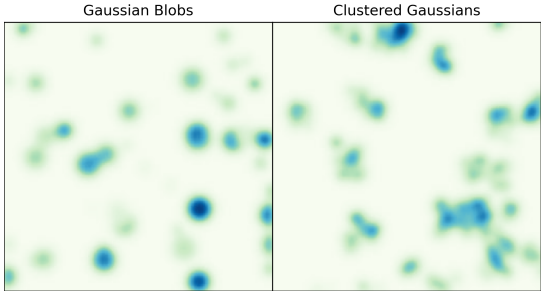

Fig. 1: Two types of data used throughout the scope of this project.

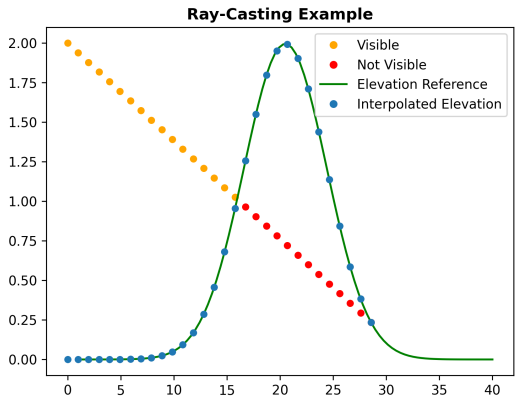

Fig. 2: Rays encountering high elevations are blocked, creating shadows or holes in the regions behind these occlusions, accurately simulating the observed incompleteness in the maps.

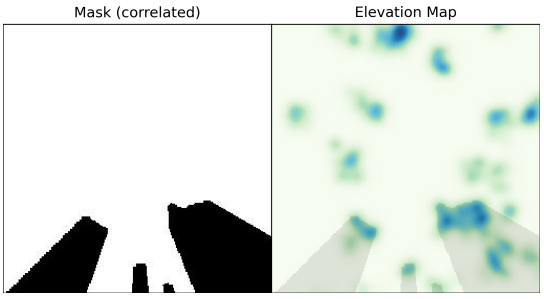

Fig. 3: The edges of the masks coincide with regions of high elevation for a sensor placed in the middle of the environment/image.

C. Masking through Ray-Casting

The original implementation of MAT employs random masks during training and validation. However, our focus is on inpainting incomplete maps where missing regions are caused by environmental occlusions. These occlusions introduce holes in the maps that are directly correlated with the occluding structures in the environment, such as regions of high elevation. To model this correlation, we utilized ray-casting technique to generate masks corresponding to these high-elevation areas. As illustrated in figure (III-C), rays encountering high elevations are blocked, creating shadows or holes in the regions behind these occlusions, accurately simulating the observed incompleteness in the maps. As seen in figures (4,6 & 7), we used three different types of masks in our study: random, 360-deg (mimicking a 360-deg LiDAR) and forward facing masks (for forward facing cameras). Figure (3) shows the correlation between a mask and the corresponding elevations. The edges of the masks coincide with regions of high elevation for a sensor placed in the middle of the environment/image.

D. Uncertainty Quantification

In safety-critical applications such as autonomous driving, the ability of a network to predict uncertainty is crucial to avoid over-reliance on overconfident predictions, which could lead to unsafe maneuvers. The style manipulation module, designed for generating pluralistic inpaintings, also facilitates uncertainty estimation in the network's predictions. By employing a noise injection mechanism, we generate an ensemble of N diverse outputs for the same masked input elevation map. The variability among these outputs enables the computation of key uncertainty metrics, including the mean, RMSE, and variance, providing a robust quantification of prediction uncertainty. We discuss these results in section V.

IV. Experiments

We conducted our experiments on two dataset types previously described, with all images uniformly sized to a resolution of 256×256 pixels. For the masking strategy, we utilized three types of masks—random, 360-degree, and forward-facing—during both training and testing. Initially, we evaluated a pre-trained model trained on the CelebHQ dataset to establish a baseline. Subsequently, we trained our own models, using the same initial hyperparameter configurations as the pre-trained model. The details of the experiments carried out are: (1) Celebs & gaussian blobs - evaluation was conducted using the pretrained model with random masks applied to the Gaussian blobs dataset. No additional training was performed. (2) Custom 1 & gaussian blobs - a custom model was trained using the same hyperparameters as the pretrained model described in Section 1. The dataset consisted of Gaussian blobs, and random masks were utilized for both training and validation. (3) Custom 2b & gaussian clusters - another custom model was trained, maintaining the same hyperparameters. The dataset was modified to include gaussian clusters instead of blobs, and correlated 360-degree masks were used during training and validation. (4) Custom 3 & gaussian clusters - this model was also custom trained with Gaussian clusters as the dataset. To simulate a forward-facing camera setup, masks corresponding to a front-facing vision cone were generated and applied during training and validation, ensuring alignment with the intended application scenario. The training took place on **insert hardware details here**.

V. Results & Discussion

TABLE I: Model Metrics

Model	LPIPS	PSNR	SSIM	L1	FID
CelebsHQ (baseline)	0.2791	19.70	0.8524	0.0536	162.12
Model 1	0.2072	20.33	0.8704	0.0436	69.20
Model 2.b	0.1892	18.76	0.8479	0.0486	103.78
Model 3	**0.0276**	**29.87**	**0.9768**	**0.0078**	**7.31**

Table (I) lists various metrics for the models that we trained. Considering the baseline was trained on a completely different dataset than that used for inference, the results are expected to be poor. This was done simply to get a point of reference for future models. Though Custom 1 was trained on our own data, it was trained using a random mask which we attribute to the poor model convergence and overall poor performance. Custom 2.b was trained on custom data and 360°. We found the 360° masks to be a little too aggressive, leaving little data on the image to infer structure from. Model 3 easily gave the best results of all the models as it used a less aggressive forward facing mask. It should be noted that in all cases the metrics are calculated over the entire image. A perhaps more fair comparison would be to compare inpainted regions only.

- **Celebs & Gaussian blobs:** The reconstructed images in figure(4) exhibit poor quality, with artifacts resembling facial features, such as wisps of hair or shapes resembling eyes. In many cases, the reconstructions included intriguing yet unintended facial-like patterns. While this approach was clearly insufficient for practical use, it provided a valuable starting point for further refinement and experimentation.

- **Custom 1 & Gaussian blobs:** Figure 5 presents the results from our first custom-trained model. Unlike earlier experiments, the reconstructions no longer exhibit unintended facial features. However, the overall reconstruction quality remains suboptimal. This is likely attributed to the random nature of the training data and the presence of large masks used during the training process. While the masks were randomly generated, many covered a significant portion of the image, negatively impacting the model's performance. These observations prompted a re-evaluation of the mask design to better align with the intended use case.
- **Custom 2b & Gaussian clusters:** The inference results presented in Figure (6) correspond to a custom-trained model using Gaussian clusters and 360-degree masks. The 360-degree masks exhibit a level of aggressiveness comparable to the random masks employed in the original pipeline. However, the evaluation metrics show improved performance compared to the CelebHQ results, as the model was specifically trained on the elevation dataset, enabling it to better handle the characteristics of this data.
- **Custom 3 & Gaussian clusters:** The final model was trained using Gaussian clusters and forward-facing masks. These masks are less aggressive compared to the 360-degree masks, as approximately half of the elevation data remained unmasked and was treated as known during training. This reduced masking intensity is reflected in the improved evaluation metrics, demonstrating better performance for this model.

A. *Uncertainty Quantification*

As previously discussed, confidence metrics are estimated by analyzing the ensemble outputs of the model. For each input image, 500 inpainted outputs are generated, enabling the calculation of key metrics such as mean, RMSE, and variance. As shown in figure 8, the variance of the inpainting is generally low. However, certain instances reveal an inconsistency where the variance remains low despite a high RMSE. This discrepancy is concerning, as it could lead to unsafe decisions in robotic applications. Addressing this issue and improving the reliability of uncertainty estimation remains a focus for future work.

VI. Conclusion

In this paper we investigated the use of a mask aware transformer [15] for inpainting elevation maps. We trained different versions of the model on different type of elevation data and mask inputs and compared their performance. In the original work which we borrowed heavily from on the model, they use a random mask generator in the training process. In the case of elevation map inpainting, the inpainting is typically due to sensor sparsity or occlusion which lead to very specific mask shapes that in the case of occlusion are correlated with the input image. As such we generated our own correlated masks through ray-casting. We found that in the case of a very aggressive mask such as a 360 degree mask you might get from a LiDAR the model converged to generating noisy images at best. However, with a less aggressive mask such as you might get from a forward facing camera assuming you have some elevation data present, we achieved very good results. We also quantified the uncertainty of our best model(model3). We found that in an ensemble of 500 inferences, the variance of results was low in almost all cases, however particularly in areas with very large masks, there was a tendency to miss potentially lethal obstacles in the inference. This could lead to dangerous behavior and must be remedied before deploying on an actual robot. We leave that for future work. Our intention is to extend this to include both elevation and semantic information. We would also like to compare these results with other probabilistic models such as Markov Chain Models which are known to be able to produce statistically similar data from a very small amount of training data.

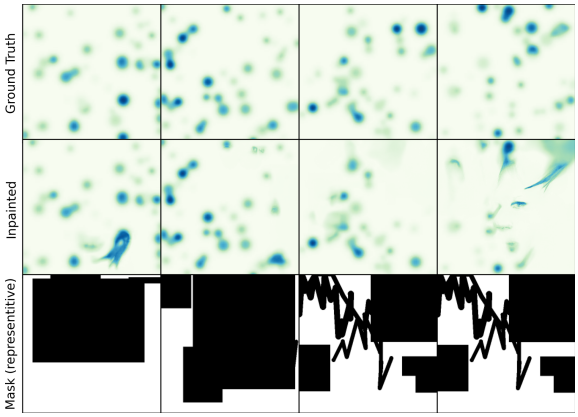

Fig. 4: Inference results on model trained on CelebHQ dataset, but tested on gaussian blobs and random masks.

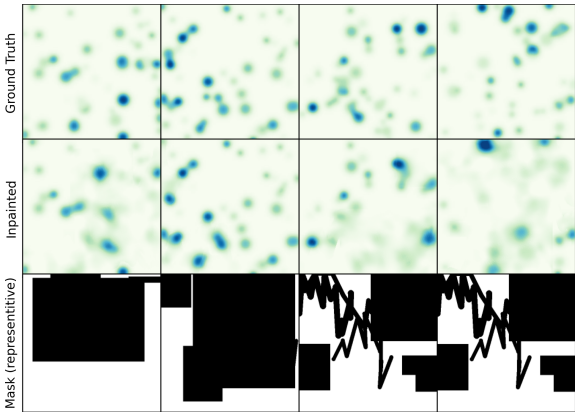

Fig. 5: Inference results on a custom trained model on gaussian blobs and random masks.

References

[1] Y. Han, J. Banfi, and M. Campbell, "Planning paths through unknown space by imagining what lies therein," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds., vol.

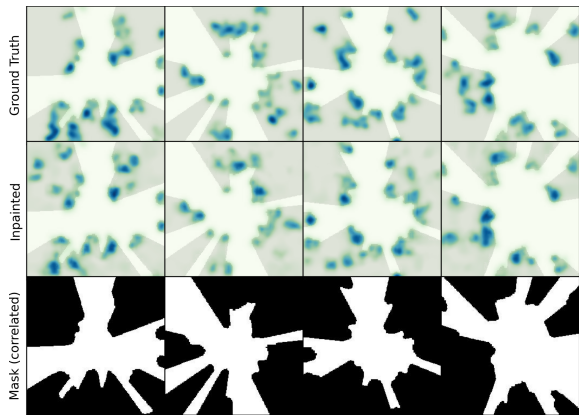

Fig. 6: Custom trained model inferences on gaussian clusters and 360-deg masks

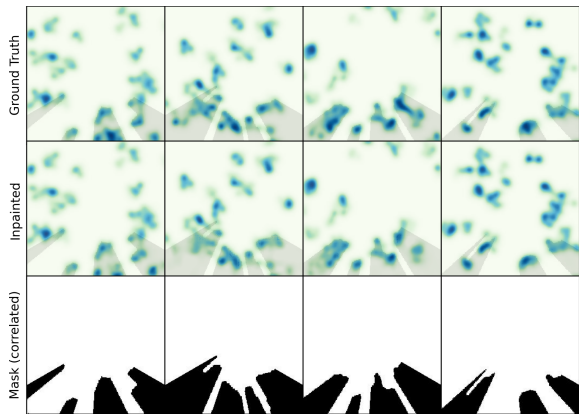

Fig. 7: Inference results on a custom trained model on gaussian clusters and forward facing masks.

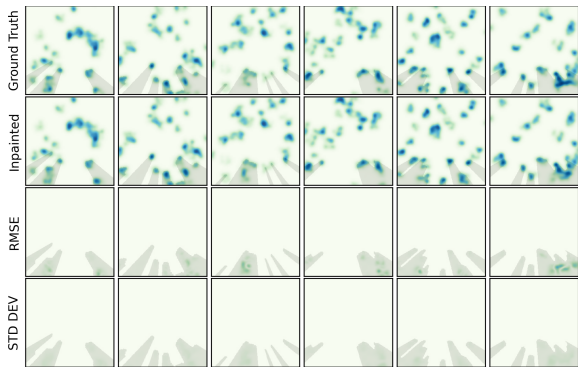

Fig. 8: Uncertainty quantification is enabled using the SMM, generating an ensemble of 500 output images for the same input image.

155. PMLR, 16–18 Nov 2021, pp. 905–914. [Online]. Available: https://proceedings.mlr.press/v155/han21a.html

[2] M. Stolzle, T. Miki, L. Gerdes, M. Azkarate, and M. Hutter, "Reconstructing occluded elevation information in terrain maps with self-supervised learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, p. 1697–1704, Apr. 2022. [Online]. Available: http://dx.doi.org/10.1109/LRA.2022.3141662

[3] L. Wang, H. Ye, Q. Wang, Y. Gao, C. Xu, and F. Gao, "Learning-based 3d occupancy prediction for autonomous navigation in occluded environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4509–4516.

[4] J. Banfi, L. Woo, and M. Campbell, "Is it worth to reason about uncertainty in occupancy grid maps during path planning?" 2022. [Online]. Available: https://arxiv.org/abs/2205.14251

[5] B. H. Wang, B. Asfora, R. Zheng, A. Peng, J. Banfi, and M. Campbell, "Multiple-hypothesis path planning with uncertain object detections," 2023. [Online]. Available: https://arxiv.org/abs/2308.07420

[6] X. Meng, N. Hatch, A. Lambert, A. Li, N. Wagener, M. Schmittle, J. Lee, W. Yuan, Z. Chen, S. Deng, G. Okopal, D. Fox, B. Boots, and A. Shaban, "Terrainnet: Visual modeling of complex terrain for high-speed, off-road navigation," 2023. [Online]. Available: https://arxiv.org/abs/2303.15771

[7] H. Wang, J. Wang, and L. Agapito, "Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 13 293–13 302.

[8] Z. Feng, H. Zhan, Z. Chen, Q. Yan, X. Xu, C. Cai, B. Li, Q. Zhu, and Y. Xu, "Naruto: Neural active reconstruction from uncertain target observations," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 21 572–21 583.

[9] S. Lee, L. Chen, J. Wang, A. Liniger, S. Kumar, and F. Yu, "Uncertainty guided policy for active robotic 3d reconstruction using neural radiance fields," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 070–12 077, 2022.

[10] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," 2020. [Online]. Available: https://arxiv.org/abs/2003.08934

[11] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," 2023. [Online]. Available: https://arxiv.org/abs/2308.04079

[12] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," 2019. [Online]. Available: https://arxiv.org/abs/1812.03828

[13] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics*, vol. 41, no. 4, p. 1–15, July 2022. [Online]. Available: http://dx.doi.org/10.1145/3528223.3530127

[14] K. Katyal, K. Popek, C. Paxton, P. Burlina, and G. D. Hager, "Uncertainty-aware occupancy map prediction using generative networks for robot navigation," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 5453–5459.

[15] W. Li, Z. Lin, K. Zhou, L. Qi, Y. Wang, and J. Jia, "Mat: Mask-aware transformer for large hole image inpainting," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 758–10 768.

APPENDIX