2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Thermal Voyager: A Comparative Study of RGB and Thermal Cameras for Night-Time Autonomous Navigation

Aditya NG, Dhruval PB, Jehan Shalabi, Shubhankar Jape, Xueji Wang and Zubin Jacob
{analgund, dpobbath, shalabi, sjape, wang4008, zjacob}@purdue.edu*

***Abstract*— Achieving reliable autonomous navigation during nighttime remains a substantial obstacle in the field of robotics. Although systems utilizing Light Detection and Ranging (LiDAR) and Radio Detection and Ranging (RADAR) enable environmental perception regardless of lighting conditions, they face significant challenges in environments with a high density of agents due to their dependence on active emissions. Cameras operating in the visible spectrum represent a quasi-passive alternative, yet they see a substantial drop in efficiency in low-light conditions, consequently hindering both scene perception and path planning. Here, we introduce a novel end-to-end navigation system, the "Thermal Voyager", which leverages infrared thermal vision to achieve true passive perception in autonomous entities. The system utilizes our architecture, TrajNet to interpret thermal visual inputs to produce desired trajectories and employs a model predictive control strategy to determine the optimal steering angles needed to actualize those trajectories. We train our TrajNet on a comprehensive video dataset incorporating visible and thermal footage alongside Controller Area Network (CAN) frames. We demonstrate that nighttime navigation facilitated by Long-Wave Infrared (LWIR) thermal cameras can rival the performance of daytime navigation systems using RGB cameras. Our work paves the way for scene perception and trajectory prediction empowered entirely by passive thermal sensing technology, heralding a new era where autonomous navigation is both feasible and reliable irrespective of the time of day. We make our code and thermal trajectory dataset public[1].**

I. INTRODUCTION

Autonomous navigation in low-light and nighttime conditions has long been a formidable challenge in the field of robotics. As we continue to rely on autonomous agents for an ever-expanding array of applications, from self-driving cars to search and rescue missions, the need for reliable navigation in adverse environments becomes increasingly apparent. Traditional visible spectrum cameras, while effective during daylight hours, falter in low-light conditions, often rendering autonomous systems incapable of safe and effective operation. In contrast, thermal vision technology offers a promising alternative, enabling autonomous agents to navigate successfully in darkness.

While visible spectrum cameras rely on ambient light, which diminishes after sunset, thermal vision operates on the principle of detecting heat emissions, allowing it to function effectively in complete darkness. This capability is particularly vital in various scenarios, such as military operations, emergency response missions, and autonomous transportation, where operating conditions often extend into the night. The LWIR regime is particularly useful for this task as the radiance of objects at room temperature peaks in the 8-14μm range. The atmospheric transmission in this wavelength band is also the highest, which mitigates the effects of environmental disturbances on the signal in this spectrum. As shown in Fig. 2, the RGB system's visibility is greatly limited by the ambient lighting and this causes a great deal of uncertainty for downstream perception and path planning systems. By harnessing thermal vision, autonomous agents can maintain their navigational competence regardless of lighting conditions, ensuring both safety and reliability.

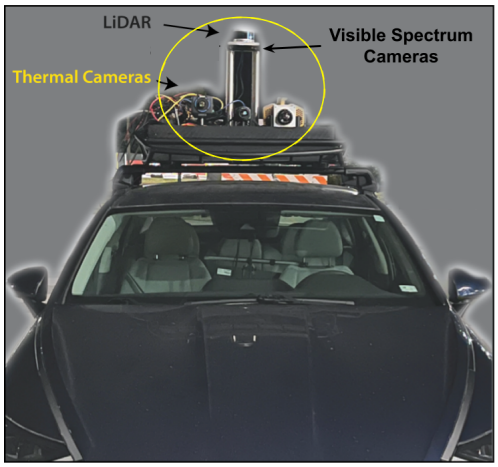

Fig. 1: The Thermal Voyager Autonomous vehicle setup: We mount our sensor stack onto a vehicle with drive-by-wire functionality

LiDAR systems rely on laser pulses that can be obstructed by fog, rain, or dust particles, rendering them less reliable in adverse weather conditions. Similarly, RADAR can struggle to detect objects with high levels of precision and detail, leading to potential navigation errors. When multiple agents perceive a scene simultaneously, active modalities such as LiDAR and RADAR become fundamentally challenging due to signal interference. In contrast, thermal vision provides a fully passive approach for scalable perception. The continuous, real-time image of the environment without the limitations posed by sporadic laser pulses or radio waves also offers a more reliable means of nighttime navigation.

Research like [1], [2], [3], [4], [5], [6] which draw inspiration from ViT [7] showcases the domain-agnostic learning strengths of the transformer. To adapt these models to new domains, we explore the significance of utilizing ther-

* Purdue University, West Lafayette, IN, USA
[1] https://adityang.github.io/TrajNet

mal vision for autonomous navigation at night, highlighting its advantages over visible spectrum cameras, LiDAR, and RADAR systems, and how it can revolutionize autonomous operations in environments where human visibility is reduced to zero. We experimentally validate that TrajNet in thermal mode at night time is equivalent to RGB mode in daytime. We propose the first end-to-end thermal navigation system which we evaluate in simulator and in the real world on our vehicle shown in Fig. 1

II. Related work

HADAR. Recent advancements in heat-assisted detection and ranging (HADAR) [8] have demonstrated the potential of thermal perception in depth perception. HADAR presents an innovative platform by integrating the capabilities of thermal imaging with artificial intelligence. It enables users to see through the darkness, delivering a visual clarity akin to broad daylight. This consistent visibility regardless of ambient lighting conditions revolutionizes various fields, from medical diagnosis to remote sensing. Moreover, while most computer vision systems are currently trained on visible image data, HADAR provides the platform to adapt existing foundational models [6], [9], [10] to the thermal vision domain.

Trajectory Prediction. *End-to-End Learning for Self-Driving Cars* [11] marked a paradigm shift by using deep neural networks to map raw pixels to steering commands. *ChauffeurNet* [12] employed imitation learning with synthesized data for trajectory prediction while *MPPI: Model Predictive Path Integral Control* [13] focuses on Model Predictive Control (MPC). All these works are limited by their reliance on visible light imagery and hence degrade in performance in low light conditions. Our TrajNet system fills this gap by utilizing thermal imaging for consistent performance regardless of lighting conditions. *End-to-End Training of Deep Visuomotor Policies* [14] simplifies the development process by learning sensory-motor mappings directly from raw visual data. However, this approach requires extensive data and computational resources and may lack interpretability. *Visual Semantic Planning using Deep Successor Representations* [15] employs imitation learning and reinforcement learning for visual planning but is limited by its reliance on a simulated physics engine, making it less applicable in real-world, unpredictable environments. *Deep Visual Teach and Repeat on Path Networks* [16] uses a single image sequence for efficient decision-making but is limited by its inability to handle complex steering and lacks error correction. *Learning to Navigate in Cities Without a Map* [17] uses Google Street View data for deep RL-based navigation. While innovative, it is constrained by the availability and quality of Street View data and doesn't address nighttime navigation challenges.

Thermal Perception. *Thermal-Inertial Localization* [18] combines thermal imaging with inertial data for navigation in smoke-filled environments but is limited to aerial robots and relies on additional sensors. TrajNet offers a more versatile solution suitable for both aerial and ground navigation. *RGB-T SLAM and Hand-held monocular SLAM in thermal-infrared* [19], [20] focus on mapping and localization by integrating thermal and visible spectrum data but do not delve into predictive capabilities, particularly in trajectory determination. *Practical Infrared Visual Odometry* [21] explores the potential of infrared cameras in visual odometry but lacks a comprehensive end-to-end solution for trajectory prediction.

While existing studies have explored various aspects of sensor-based navigation and thermal perception, TrajNet distinguishes itself by integrating thermal vision into an end-to-end solution tailored for reliable nighttime navigation.

III. Proposed Work

A. Vehicle Setup

Thermal Camera. Thermal data acquisition was executed employing a FLIR A325sc, a Long Wave Infrared (LWIR) thermal camera equipped with an uncooled vanadium oxide microbolometer detector. The camera's specifications encompass a resolution of 320 x 240 pixels, a detector pitch measuring 25μm, a time constant of 12 ms, a focal length of 18 mm, and an f-number of 1.3 with a horizontal FoV of 45^o and a vertical FoV of 33.8^o.

RGB Camera. The ZED 2i stereo camera from STEREOLABS was used as the RGB camera in the experiments. The resolution of the camera is 1920 x 1080 pixels, 70^o FOV, and 2 microns pixel size.

Compute Platform. The compute platform used in our experiments was an NVIDIA Jetson AGX Orin with 2048 CUDA cores, 64 tensor cores, 32 GB RAM with a 12-core Arm® Cortex®-A78AE v8.2 64-bit CPU. Model training was performed on an NVIDIA RTX A6000 GPU with 48GB VRAM using PyTorch [23].

B. System architecture

The architecture of the proposed system, as delineated in Fig. 5, amalgamates an end-to-end differentiable neural network exclusively relying on visual input with MPC [24] to transform designated trajectories into actuation commands. TrajNet ingests camera frames in either the LWIR or the visible spectrum as its input modality. The architecture employs a ResNet [9] or SWIN DPT backbone [6], [10] for feature extraction, culminating in a feature tensor. These features either regress into a trajectory tensor or are employed to generate a one-hot encoded vector for template trajectory selection. The generated trajectory is input into MPC, which optimizes a sequence of control inputs over an immediate temporal horizon of approximately 1 second (corresponding to 50 discrete steps). The computational latency for the entire pipeline, including both the neural network and MPC, is approximately 50 ms. Control sequences are cached and periodically transmitted to the low-level controller, with a cache validity threshold of 70 ms. Exceeding this duration triggers a disengagement protocol in the TrajNet, accompanied by user notification.

Fig. 2: Navigating in dark environments using RGB cameras can be challenging due to the lack of features. Thermal vision provides a truly passive approach to environmental perception.

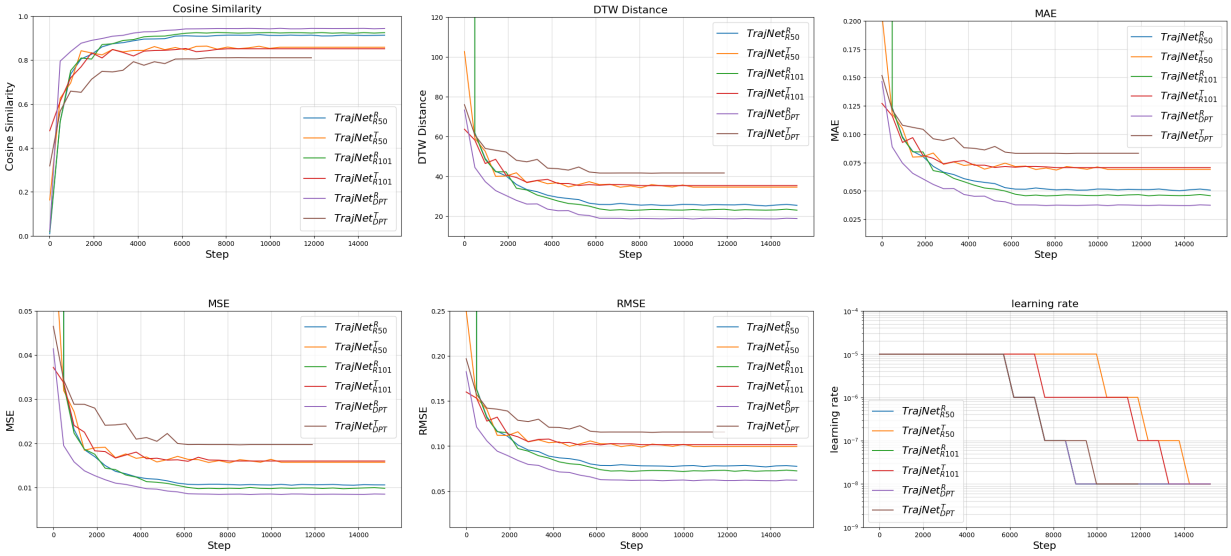

Fig. 3: We train 6 different architectures for 15 epochs and have visualized the metrics and the learning rate as the model learns. The Cosine Similarity metric increases and the DTW Distance, MAE, MSE, and RMSE decrease through the course of training. The learning rate reduces in stages and is governed by the $ReduceLROnPlateau$ scheduler. We make use of the Adam optimizer [22] with β set to $(0.9, 0.999)$ and ϵ value of 10^{-8}

C. Safety Considerations

Ensuring operational safety is paramount in the context of autonomous vehicle development. Accordingly, a rigorous Failure Modes and Effects Analysis (FMEA) has been conducted on all constituent subsystems. Based on this analysis, the TrajNet operates under a set of predefined safety constraints: (1) the engagement button is only enabled when the vehicle is at a standstill with the driver's foot on the brake; (2) disengagement is triggered when the vehicle speed exceeds 5 MPH or upon manual driver intervention involving the throttle, brake, or steering systems; (3) auditory alerts are triggered during transitions between engagement and disengagement states; (4) TrajNet is restricted to a maximum of 20% throttle actuation; and (5) system shutdown is ensured upon powering off the vehicle. This robust safety framework facilitates rigorous exploration within the domain of autonomous navigation under low-illumination conditions.

D. Data Augmentation

In order to increase the variety of scenes our model is exposed to during training, we augment our dataset.

2D to 3D transformations. We make use of a function to convert between an affine transform on the 2D camera plane and the corresponding 3D affine transform in the world coordinate system given by $M_{2D} \longleftrightarrow M_{3D}$. Consider a 2D affine transform M_{2D} and the intrinsic matrix K of the camera. If we apply a 2D affine transformation M_{2D} on the image plane, it means we're changing the 2D image coordinates. The equivalent 3D transformation M_{3D} corresponding to the 2D affine transformation M_{2D} on the image plane can be obtained from Eq. (1). Here, K^{-1} represents the pseudo-inverse of the matrix K.

$$M_{3D} = K^{-1} M_{2D} K \tag{1}$$

Inversion We can flip the image and the trajectory along the y-axis to generate a new data point. We set M_{2D} as shown inEq. (2). Using Eq. (1), we can compute the corresponding 3D affine transform to be as shown in Eq. (2).

$$M_{2D}^{flip} = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & -1.0 & 2c_y \\ 0.0 & 0.0 & 1.0 \end{bmatrix}; M_{3D}^{flip} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

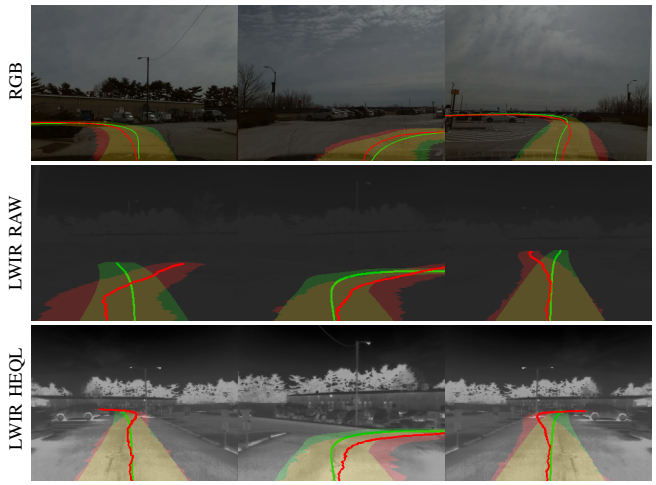

Fig. 4: Visuals of TrajNet working on the validation set of our dataset. The trajectory in green is the ground truth trajectory, red is the model prediction and the yellow regions represent the overlap. The two trajectories have a bright green and red line running through them which indicates their center. The model's performance on histogram-equalized LWIR tends to be better than raw LWIR.

Random Skew We apply a random skew on the image plane along the y-axis as defined by the 2D affine transform M_{2D} in Eq. (3). This skew is limited to be between -0.1 and 0.1 and its effect is visually described in Fig. 6. The 3D skew transformation in terms of the given parameter warp_y and the intrinsic parameters f_x, f_y, c_x, c_y can be represented as shown in Eq. (4)

$$M_{2D}^{skew} = \begin{bmatrix} 1.0 & warp_y & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \tag{3}$$

$$M_{3D}^{skew} = \begin{bmatrix} 1.0 & \text{warp}_y & \frac{c_x \times \text{warp}_y - c_x}{f_x} \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \tag{4}$$

E. Vehicle Model

$$\dot{x} = v\cos(\theta+\beta); \dot{y} = v\sin(\theta+\beta); \dot{\theta} = \frac{v\cos(\beta)\tan(\delta)}{L} \tag{5}$$

We make use of the Bicycle Vehicle Model. The parameters of the model are as follows: (x, y) are the coordinates of the center of gravity (CG) of the bicycle model, θ is the heading angle, v is the velocity of the center of gravity of the bicycle model, δ is the steering angle, L is the wheelbase (distance between front and rear axle), L_r is the distance from the CG to the rear axle, L_f is the distance from the CG to the front axle, $\beta = \arctan\left(\frac{L_r \cdot \tan(\delta)}{L}\right)$ is the slip angle at the vehicle's CG. The dynamic equations of the bicycle model are shown in Eq. (5). Here, $\dot{x}$, $\dot{y}$, $\dot{\theta}$, and $\dot{v}$ represent the time derivatives of x, y, θ, and v, respectively.

Algorithm 1: Warping Simulator

Input: Neural Network N,
Image Frame $F \in \mathbb{R}^{1\times C\times H\times W}$,
Ground Truth Trajectory $\mathbf{T}_{\text{gt}} \in \mathbb{R}^{250\times 2}$,
Maximum Distance d_{max},
Maximum Iterations n_{max},
Intrinsic Matrix $\mathbf{K} \in \mathbb{R}^{3\times 3}$,
Distortion Coefficients $\mathbf{D} \in \mathbb{R}^5$

Output: Loss $\mathcal{L}$

$\mathcal{L} \leftarrow 0.0$
$d_{\text{travelled}} \leftarrow 0.0$
iter_count $\leftarrow 0$
$\mathbf{M}_{\text{3D}} \leftarrow I_4$
$\mathbf{M}_{\text{2D}} \leftarrow I_3$

while $d_{travelled} < d_{max}$ ***and*** *iter_count* $< n_{max}$ **do**

$\mathbf{T}_{\text{gt}} \leftarrow \mathbf{M}_{\text{3D}}[:3,:3]\mathbf{T}_{\text{gt}}^{\top} + \mathbf{M}_{\text{3D}}[:3,3]$
$F' \leftarrow \text{WarpAffine2D}(F, \mathbf{M}_{\text{2D}})$
$\mathbf{T}_{\text{pred}} \leftarrow N(F')$
$\mathbf{p}_{\text{3D}} \leftarrow \mathbf{T}_{\text{pred}}[1]$
$d_{\text{travelled}} \leftarrow d_{\text{travelled}} + \|\mathbf{p}_{\text{3D}}\|$
$\mathbf{M}_{\text{3D}}^{\text{new}} \leftarrow \begin{pmatrix} 1 & 0 & 0 & \mathbf{p}_{\text{3D}}[0] \\ 0 & 1 & 0 & \mathbf{p}_{\text{3D}}[1] \\ 0 & 0 & 1 & \mathbf{p}_{\text{3D}}[2] \\ 0 & 0 & 0 & 1 \end{pmatrix}$
$\mathbf{M}_{\text{3D}} \leftarrow \mathbf{M}_{\text{3D}} \times \mathbf{M}_{\text{3D}}^{\text{new}}$
$\mathbf{M}_{\text{2D}} \leftarrow \text{Transform3Dto2D}(\mathbf{M}_{\text{3D}}, \mathbf{K})$
$\mathcal{L} \leftarrow \mathcal{L} + \text{MSE}(\mathbf{T}_{\text{pred}}, \mathbf{T}_{\text{gt}})$
iter_count $\leftarrow$ iter_count $+ 1$

end
return $\mathcal{L}$

F. Generating Trajectory Labels

We gather CAN data from the vehicle which includes steering angle and wheel speed data per frame. This is gathered at a frequency of about 30 Hz. Using the vehicle model shown in Eq. (5), we can convert a set of N consecutive timestamped frames containing steering angle and wheel speed into a trajectory of N points. These generated trajectory labels are grouped into sets of 3 seconds. These sets are used to supervise our trajectory network. The trajectory is visualized by projecting it onto the camera plane using the camera intrinsic matrix as shown in Fig. 4.

G. Model Predictive Control

Once $TrajNet$ produces a trajectory, we must then compute the optimal steering angle to execute that trajectory. We look into Model Predictive Control [25], [26], [24] and use the Bicycle Model to optimize the cost function $J(\mathbf{u})$ as described in Eq. (6) where $\mathbf{u} = [u_0, u_1, \ldots, u_{N-1}]$ is the sequence of control inputs, X_{des} is the desired trajectory, and K is the tuning parameter for the control input. The optimization and its bounds are described in Eq. (7). By optimizing Eq. (7), we get the optimal control input sequence $\mathbf{u}^*$ that minimizes the difference between the trajectory computed by the controller and the desired trajectory requested

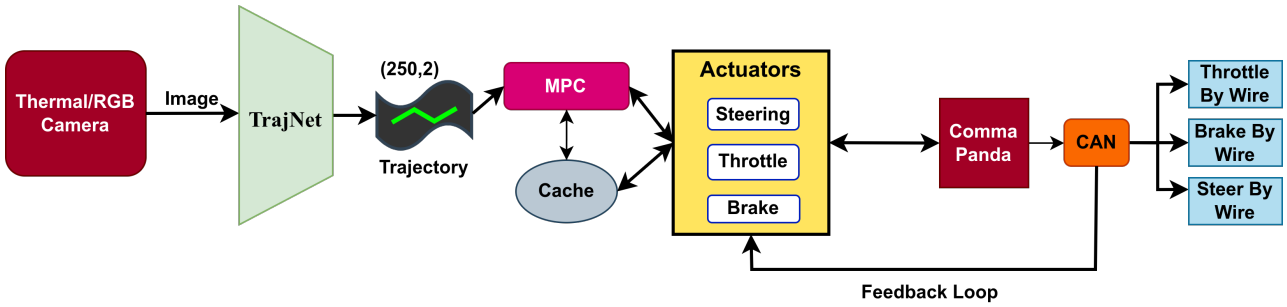

Fig. 5: **Architecture**. Thermal/RGB image frames are fed into $TrajNet$ which outputs a trajectory. The Model Predictive Controller translates the trajectory to an optimal control sequence of actuator values. Low-level systems command the actuators to execute the desired trajectory

(a) Input image (b) Skewed image

Fig. 6: An example of the skew (0.1) we use to augment our dataset and use in our simulator.

by the model.

$$J(\mathbf{u}) = \sum_{i=0}^{N-1} \left((x - X^x_{\text{des},i})^2 + (y - X^y_{\text{des},i})^2 + Ku_i^2 \right) \quad (6)$$

$$\begin{aligned} \min_{\mathbf{u}} \quad & J(\mathbf{u}) \\ \text{s.t.} \quad & u_i \in (-S_{\max}, S_{\max}) \quad \forall i \in [0, N-1] \end{aligned} \quad (7)$$

H. Simulator

Although TrajNet correctly predicts the ground truth trajectory, when taken into the real world, the initial training isn't sufficient to teach it to stay on track and correct large track deviations well. The real-world noises that come from factors such as wind, steering lag, and small undulations on the road cause deviations. The training data which consists of human driving doesn't necessarily teach the model to recover from specific deviations. Hence, if the car deviates from the expected trajectory, it does not predict a trajectory to recover and thus continues to drift from the ideal trajectory. To address this issue, we take inspiration from [28] and introduce a warping simulator as described in Algo. 1 that uses the $M_{2D} \longleftrightarrow M_{3D}$ transform described in Eq. (1) to mimic the effects of driving. As the vehicle moves forward in the simulator, we warp the image by the corresponding amount to produce the illusion of movement. Fine-tuning the network on this simulator shows improvement in the model's ability to recover from track deviations. We avoided conventional simulators including AirSim [29] and Carla [30] due to their limited support for thermal simulation at the time and went with our custom simulator.

IV. Experiments

We gather a training dataset consisting of about 45,000 frames where each frame consists of an Image and Trajectory pair as shown in Fig. 4. This dataset was gathered in an empty parking lot along a desired track. We trained a total of 6 different architectures and experimented with three backbones: $ResNet_{50}$, $ResNet_{101}$ [9] and $DPT_{SWIN-2T}$ [6], [10]. We took 2 approaches: trajectory regression and template selection. In trajectory regression, the features extracted from the backbone go through convolution and fully connected layers to produce the trajectory. The regression mode permits a large amount of flexibility in what the model can predict, but it also means that the model could output invalid trajectories (which cannot be executed with the constraints of the vehicle model); and since every trajectory can be different, we would need to run our MPC online and adds latency. In the case of templates, the model produces a one-hot encoding to select from a set of template trajectories. The template approach removes the possibility of invalid trajectories and reduces the time spent running MPC online as all the possible trajectories are known and the corresponding control sequence can be pre-computed. The inherent limitation in employing a template-based methodology lies in the model's restricted ability to adapt and assimilate novel trajectories. Attempting to instruct the model using an extensive dataset of trajectories becomes ineffective as the model cannot learn to discern minute distinctions among a large number of trajectories. We first train our models on the visible spectrum dataset as a baseline and then fine-tune them to operate on the LWIR image space. We experiment with providing the raw LWIR input as well as the histogram equalized LWIR input.

A. Ablation Study

We observe from Tab. I and Fig. 3 that the $TrajNet^R_{DPT}$ preforms the best overall. $TrajNet^R_{R101}$ seconds it in terms of trajectory accuracy, but cannot operate in real-time.

Method	Dataset	Hyperparameters				Metrics				
		RM	BS	EP	LR	Cos Sim	DTW Distance	MAE	MSE	RMSE
$TrajNet^{R}_{R50}$	RGB	$R_{250\times2}$	60	15	0.00001	0.9138^{*}	25.2496^{*}	0.0505^{*}	0.0105^{*}	0.0771^{*}
	LWIR RAW	$R_{250\times2}$	60	15	0.000001	0.6213	55.2040	0.1104	0.0188	0.1313
	LWIR HEQL	$R_{250\times2}$	60	15	0.000001	0.6597^{+}	53.8297^{+}	0.1077^{+}	0.0173^{+}	0.1265^{+}
$TrajNet^{T}_{R50}$	RGB	T_{10}	60	15	0.00001	0.8540	35.1748	0.0703	0.0160	0.1009
	LWIR RAW	T_{10}	60	15	0.000001	0.8192	32.0589	0.0651	0.0072	0.0721
	LWIR HEQL	T_{10}	60	15	0.000001	0.8890^{*+}	30.0679^{*+}	0.0612^{*+}	0.0069^{*+}	0.0712^{*+}
$TrajNet^{R}_{R101}$	RGB	$R_{250\times2}$	60	15	0.00001	0.9241^{*}	23.2562^{*}	0.0465^{*}	0.0098^{*}	0.0728^{*}
	LWIR RAW	$R_{250\times2}$	60	15	0.000001	0.9110	26.3298	0.0527^{+}	0.0062	0.0643^{+}
	LWIR HEQL	$R_{250\times2}$	60	15	0.000001	0.9141^{+}	26.5019^{+}	0.0530	0.0058^{+}	0.0646
$TrajNet^{T}_{R101}$	RGB	T_{15}	60	15	0.00001	0.8510	35.2925	0.0706	0.0160	0.1016
	LWIR RAW	T_{15}	60	15	0.000001	0.8255	33.4171	0.0668	0.0110	0.0860
	LWIR HEQL	T_{15}	60	15	0.000001	0.8992^{*+}	30.0589^{*+}	0.0601^{*+}	0.0069^{*+}	0.0701^{*+}
$TrajNet^{R}_{DPT}$	RGB	$R_{250\times2}$	30	15	0.00001	0.9426^{*}	18.7051^{*}	0.0374^{*}	0.0085^{*}	0.0621^{*}
	LWIR RAW	$R_{250\times2}$	30	15	0.000001	0.4079	63.8079	0.1276	0.0238	0.1482
	LWIR HEQL	$R_{250\times2}$	30	15	0.000001	0.8247^{+}	39.3470^{+}	0.0787^{+}	0.0109^{+}	0.0955^{+}
$TrajNet^{T}_{DPT}$	RGB	T_{25}	30	15	0.00001	0.8101^{*}	41.6146	0.0832	0.0197^{*}	0.1156
	LWIR RAW	T_{25}	30	15	0.000001	0.7621	41.1822^{*+}	0.0824^{*+}	0.0217	0.1147^{*+}
	LWIR HEQL	T_{25}	30	15	0.000001	0.8001^{+}	41.9756	0.0877	0.0201^{*+}	0.1238

TABLE I: **Ablation Study** We run hyper-parameter sweeps on six different architectures to find the best-performing models for each architecture. Models with superscripts R and T use regression and template selection respectively. The subscript in T_N indicates that the model uses N trajectory templates. We sweep across the parameters: Regression Method (RM), Batch Size (BS), number of Epochs (EP) and Learning Rate (LR). We evaluate on Cosine Similarity, Dynamic Time Warping (DTW) Distance [27], Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). The best runs in $daytime^{*}$ and $nighttime^{+}$ are highlighted. Note that RGB networks are evaluated exclusively on the daytime data due to blank images seen at nighttime.

$TrajNet^{R}_{R50}$ performed comparably to $TrajNet^{R}_{R101}$, while being smaller and faster. After these investigations, it is observed that template-based networks exhibit a comparatively lower level of performance compared to regression networks. However, the deficiency in accuracy is offset by a notable improvement in computational speed, attributed to the elimination of the downstream necessity for online MPC. While we see promise in the application of the template-style architecture, we acknowledge the fact that the real world can present our systems with a diverse set of scenarios that the templates may not accommodate. We also observe that training the models on the histogram equalized thermal data produces superior results when compared to training them on the raw thermal signal. This intuitively makes sense because the histogram equalization gives more contrast to the image making it easier to perceive the track.

B. On Track Testing

After simulator validation and all the in-vehicle safety checks, the system was tested in a parking lot to drive in a circuit. The RGB system was able to drive well during the daytime with visible spectrum input, but the performance of these networks deteriorated at later times of the day as the ambient lighting went down. The thermal networks performed consistently regardless of ambient lighting.

V. Conclusion

Our research introduced TrajNet, the first end-to-end navigation system designed to utilize thermal cameras for autonomous vehicle navigation in low-light and night-time conditions. The study provides a comprehensive comparison between thermal LWIR cameras and traditional visible spectrum sensors, demonstrating the advantages of thermal imaging for passive perception agnostic to ambient lighting in autonomous vehicles. Our results show that TrajNet, when equipped with thermal cameras, exhibits robust performance even in challenging lighting conditions, overcoming the limitations of cameras operating in the visible spectrum. The thermal cameras were able to provide a dense depiction of the scene across the thermal spectrum, enabling more accurate and stable trajectory predictions. We also introduced a novel dataset comprising RGB, thermal (LWIR), and CAN frames. This dataset proved to be invaluable for training TrajNet and could serve as a foundation for future research in this domain. The promising results from our thermal camera-based system suggest an alternative pathway for the development of truly passive perception systems in autonomous vehicles that work in all weather conditions.

ACKNOWLEDGMENT

We thank Fanglin Bao and Shree Hari Sureshbabu for the fruitful discussions through the course of the project. We acknowledge funding from NSF under grant no. DMR-1654676.

REFERENCES

[1] H. Bao, L. Dong, S. Piao, and F. Wei, "BEit: BERT pre-training of image transformers," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=p-BhZSz59o4

[2] B. Graham, A. El-Nouby, H. Touvron, P. Stock, A. Joulin, H. Jegou, and M. Douze, "Levit: A vision transformer in convnet's clothing for faster inference," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 12 259–12 269.

[3] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, A. Brock, E. Shelhamer, O. Hénaff, M. M. Botvinick, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver io: A general architecture for structured inputs & outputs," 2021.

[4] J. Li, X. Xia, W. Li, H. Li, X. Wang, X. Xiao, R. Wang, M. Zheng, and X. Pan, "Next-vit: Next generation vision transformer for efficient deployment in realistic industrial scenarios," *arXiv preprint arXiv:2207.05501*, 2022.

[5] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, and B. Guo, "Swin transformer v2: Scaling up capacity and resolution," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[6] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[8] F. Bao, X. Wang, S. H. Sureshbabu, G. Sreekumar, L. Yang, V. Aggarwal, V. N. Boddeti, and Z. Jacob, "Heat-assisted detection and ranging," *Nature*, vol. 619, no. 7971, pp. 743–748, 2023. [Online]. Available: https://doi.org/10.1038/s41586-023-06174-6

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[10] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[11] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," 2016. [Online]. Available: https://developer.nvidia.com/blog/deep-learning-self-driving-cars/

[12] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," 06 2019.

[13] I. S. Mohamed, G. Allibert, and P. Martinet, "Model predictive path integral control framework for partially observable navigation: A quadrotor case study," in *16th International Conference on Control, Automation, Robotics and Vision*, Shenzhen, China, Dec 2020, ffhal-02545951v2f.

[14] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *J. Mach. Learn. Res.*, vol. 17, no. 1, p. 1334–1373, jan 2016.

[15] Y. Zhu, D. Gordon, E. Kolve, D. Fox, L. Fei-Fei, A. Gupta, R. Mottaghi, and A. Farhadi, "Visual semantic planning using deep successor representations," 10 2017, pp. 483–492.

[16] T. Swedish and R. Raskar, "Deep visual teach and repeat on path networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 1614–161 409.

[17] P. Mirowski, M. K. Grimes, M. Malinowski, K. M. Hermann, K. Anderson, D. Teplyashin, K. Simonyan, K. Kavukcuoglu, A. Zisserman, and R. Hadsell, "Learning to navigate in cities without a map," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 2424–2435.

[18] C. Papachristos, F. Mascarich, and K. Alexis, "Thermal-inertial localization for autonomous navigation of aerial robots through obscurants," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018, pp. 394–399.

[19] S. Vidas and S. Sridharan, "Hand-held monocular slam in thermal-infrared," in *2012 12th International Conference on Control Automation Robotics & Vision (ICARCV)*, 2012, pp. 859–864.

[20] L. Chen, L. Sun, T. Yang, L. Fan, K. Huang, and Z. Xuanyuan, "Rgb-t slam: A flexible slam framework by combining appearance and thermal information," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5682–5687.

[21] P. V. K. Borges and S. Vidas, "Practical infrared visual odometry," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2205–2213, 2016.

[22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[24] A. Bemporad, "Model predictive control design: New trends and tools," in *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, pp. 6678–6683.

[25] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3363–3370, 2019.

[26] D. Lam, C. Manzie, and M. Good, "Model predictive contouring control," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 6137–6142.

[27] *Dynamic Time Warping*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 69–84. [Online]. Available: https://doi.org/10.1007/978-3-540-74048-3_4

[28] H. Schäfer and G. Hotz. (2021) End-to-end lateral planning. Comma AI. Blog Post. [Online]. Available: https://blog.comma.ai/end-to-end-lateral-planning/

[29] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065

[30] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.