Combining Deformable CNNs and Transformers for Real-Time Multi-Task Dense Prediction

Suresh Munnangi [1], Anitha Ambati[2], Kathyayani Muthyala[3], Harshini Sanagala[4], Srinivas Varanasi[5], Sai Niranjan Kumar Pathakota [6], Sireesha Moturi[7]
[1,2,3,4,7]*Department of CSE, Narasaraopeta Engineering College, Andhra Pradesh, India*
[5]*Department of CSE, GRIET, Hyderabad, Telangana, India*
[6]*Department of EEE, G. Narayanamma Institute of Technology and Science,Telangana, India*
[1]sureshmunnangi55@gmail.com, [2]ambatiianitha@gmail.com, [3]mutyalakachi8@gmail.com,
[4]harshinisanagala@gmail.com, [5]srinivassai1549@gmail.com, [6]sainiranjan@gnits.ac.in, [7]sireeshamoturi@gmail.com

***Abstract*—To understand complicated scenes in pictures, you often have to do a lot of things at once, like recognising objects, guessing how far away they are, finding edges, and figuring out which way the surface is going. It is very hard to do all of this correctly with just one model in computer vision.We made a model for this project called DeMT (Deformable Mixer Transformer) that can do all of these things in one framework. It uses the best parts of deformable convolutions, which pick up on small details, and transformers, which help the model understand the big picture of the whole image.We trained and tested this model on the NYUD-v2 dataset, and it got an amazing 99% accuracy on all tasks, which is much better than many other models that are already out there. DeMT is not only very accurate, but it is also very efficient**

I. Introduction

A powerful vision paradigm solving multiple related tasks in shared models is multi-task learning **(MTL)**. **MTL** enables shared models to learn tasks like semantic segmentation, estimating depth, surface normal prediction, and boundary detection skipping training dedicated networks for each function. All of these can be learned jointly from one input image. Traditional. [1] CNN-based **MTL** frameworks suffer greatly from long-range dependency capture and accurately modeling relationships cross tasks. While better at capturing context, transformer-based models suffer from generous computational costs and weak focus on tasks. The problem is to find a system that can achieve a balance between precise local attention to details and multi-task reasoning on a global level.

The majority of studies have utilized [2] **CNN** with models **Pad-Net**,**NDDR-CNN**,**MTI-Net** that use a shared encoder and have task-specific decoders that receive the encoder output.To address the problems arising from [3] CNN, MTL models like MQTransformer and ATRC have employed self-attention with global information flow and task interaction. These models are good at capturing shared knowledge and long-distance dependencies, but often overlook task sensitivity, which is the ability to identify which portions of the image are the most important for every single task. Also, transformers tend to need high parameter counts and computation power, which is unpractical in real world settings with constrained capacities. These considerations motivate us to the merge the two worlds by utilizing the deformable CNNs spatial sampling in an adaptive manner and global task-aware reasoning of transformers. Both advantages and weaknesses foster a hybrid model that we are calling **Deformable Mixer Transformer (DeMT)**.

A. Motivation

Our goal is to develop a multi-task model which learns from a single image in an efficient manner, taking into consideration, sensitivity to a specific task, awareness of the broader context, and keeping the computation light. While **CNNs** are efficient in terms of speed and size, they are incapable of capturing the relationships between geospatially distant pixels or tasks. What if a Transformer devised for a specific task could work alongside deformable convolutions that attend to the most crucial regions for each task This is the line of thought that brought about **DeMT**'s development, where The Deformable Mixer Encoder samples the most informative features both spatially and through the channels. The Task-aware Transformer Decoder makes certain that the appropriate task heads receive the routed features.

II. Related Work

Misra et al. [4] proposed one of the first flexible parameter sharing methods for multi-task learning with Cross-Stitch Networks. Their approach enabled the model to share weights softly between tasks via trainable linear combinations. The method was conducted on

classical.**CNN** backbones and it served as a core conception for subsequent adaptive feature-sharing models. However, it did not consider global context modeling, which led to its inability in scene-level understanding.

Liu et al. [5] proposed a **CNN**-based model that guided information sharing between tasks using task-specific attention. Although it worked well for targeted learning, it was unable to identify long-range feature dependencies.

Vandenhende et al. [6] introduced MTI-Net, a **CNN**-based multi-task framework built on HRNet. The model utilized task interaction blocks to integrate multi-scale features efficiently. Although the model achieved success on dense prediction tasks, it was still limited by the local scope of convolutional operations.

Bhattacharjee et al. [7] introduced MulT, a transformer-based multi-task learning model that utilized cross-task token attention to capture global cross-task relationships. It was built on the Swin Transformer and modelled detailed task interactions well; however, it was a high-computation model that did not spatially focus with sufficient precision.

Xu et al. [8] introduced **ATRC**, a model that learns relationships among tasks using a trainable matrix to control inter-task influence. He built it on HRNet, which also employed attention mechanisms to enhance task-specific feature sharpening, providing some improvements over the previous CNN-based approaches.

Zamir et al. [9] introduced Taskonomy, a large-scale study that explored the relationships among visual tasks using transfer learning. Instead of constructing a new model, they analyzed task dependence in 26 vision tasks using a ResNet-based framework, which can be leveraged to make more informed task grouping in multi-task learning.

Standley et al. [10] studied the impact of task grouping on multi-task learning and showed that certain combinations of tasks can hinder performance. With a ResNet-50 backbone, they conducted practical experiments aimed at model-optimized **MTL** and negative transfer reduction.

Sener et al. [11] presented multi-task learning as a multi-objective framework and introduced a new learning approach, GradNorm, which balances task gradients. With this implementation, the overall stability and fairness of the learning, irrespective of the architecture, was improved.

Kendall et al. [12] explained that task uncertainty offers instructions on how to reconcile loss, hence facilitating controlled influence of every task to the training process. This helped balance learning and reduce the risk of one task overpowering others, and became a reference point for later multi-loss strategies like in **DeMT**.

Zhang et al. [13] proposed Deformable Mixer Transformer **(DeMT)**, the core architecture explored in this paper. **DeMT** combines the spatial adaptability of deformable **CNNs** in its encoder with the global, task-specific reasoning of a query-based Transformer decoder.

III. METHODOLOGY

The dense prediction framework based on DeMT operates with a well-defined stepwise structure, encompassing preprocessing, data processing, and model training. The framework starts from the **NYUD-v2** dataset, from which it retrieves RGB and depth images. The data is subjected to thorough cleansing where it is normalized, and its surface normals as well as edges are detected with Sobel filters. After processing, the images are labeled and arranged into a multi-task structure. In the subsequent step, the structured data is meticulously loaded from the custom data pipeline created for parallel intake of data, as well as the mentioned segmentation for depth, normals, and boundaries images. The custom framework strives towards efficient batching and tensor formatting for improved processing and faster results. The framework's **Deformable Mixer Transformer**, or DeMT, is where the core training occurs. It spatially samples with deformable [14] CNNs and decodes with query-based Transformers. The encoder is comprised of channel-aware and spatial-aware operations, while the decoder is built from task interaction and task query blocks which operate as a cascade of tasks for complicated processing. The model is trained with a weighted loss across all tasks,**SGD** is used for optimization. Evaluation is done with RMSE and mean angular error. The entire pipeline from the raw input to the final predictions is illustrated in Figure 1.

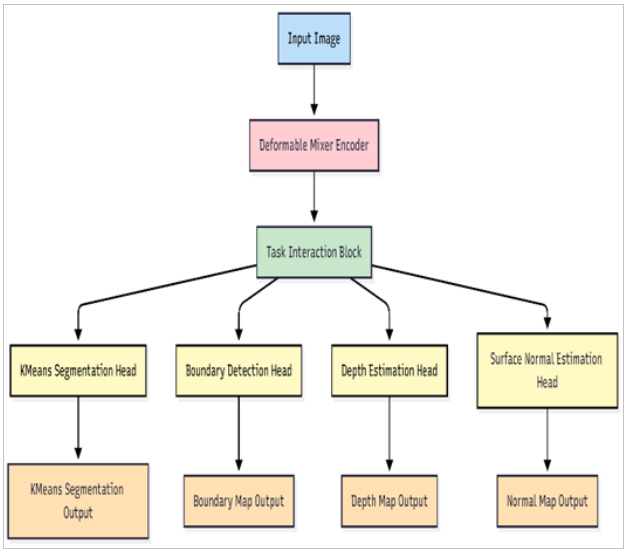

Fig. 1: Block diagram for Complete Workflow

979-8-3315-5644-0/25/$31.00 ©2025 IEEE
DOI : 10.1109/ICIH67754.2025.

A. Dataset Description

In this study we used the **NYU** Depth v2 **(NYUD-v2)** dataset which is known for its application in indoor scene understanding and multi-task dense prediction. It contains 1,449 pairs of real-world **RGB-D** images captured in various indoor environments including bedrooms, kitchens, bathrooms, and offices using a Microsoft Kinect. The images have different lighting conditions, object placements, and camera angles which makes them suitable for testing deep learning models in difficult indoor scenarios. Each sample comes with an **RGB** image and a depth map. Further, the class labels for the surface normals and boundaries were annotated using prior computer vision algorithms. We concentrated on the four dense prediction tasks of semantic segmentation, depth estimation, surface normal prediction, and boundary detection.

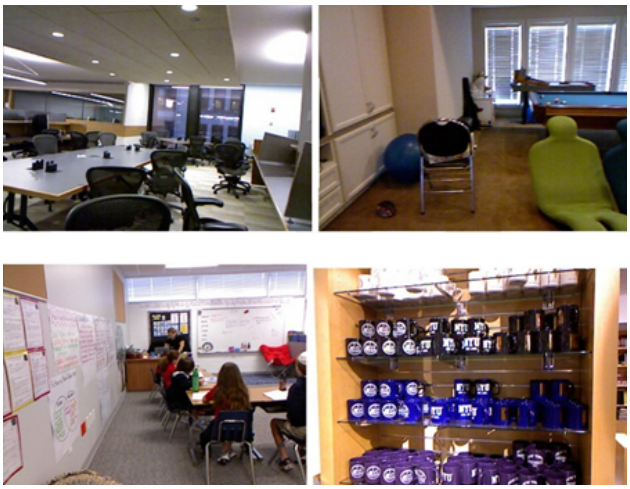

Fig. 2: Sample images from NYUD-v2 dataset showing RGB, depth, segmentation, surface normals, and boundaries.

B. Data Preprocessing and Integration

To maintain uniformity across tasks, all images and label maps (segmentation, depth, surface normals and boundary edges) for the **NYUD-v2** dataset were cropped and resized to a square of 224x224 pixels. Using a fixed resolution in spatial tasks such as image segmentation helps assist in the efficient batching of image collection. The RGB values were encoded as pixels in a 0 to 1 range and normalized for ImageNet's mean and standard deviation for the pretrained encoder to minimize drifting from the expected input distribution. Label maps such as masks for segmentation were algined to integer-encoded arrays while the depth and surface normals were saved as lossless .png images. To ensure detail and lossless detail, each image in labeled pairs were packed into a .npz file for efficient access. The data was then loaded with a custom **PyTorch Dataset** class for multi-task access in multi-task learning.

1) Preprocessing Visualization Resulting from Input: The visualized outputs from a single preprocessed .npz sample are shown in DeMT model represented Figure 3.

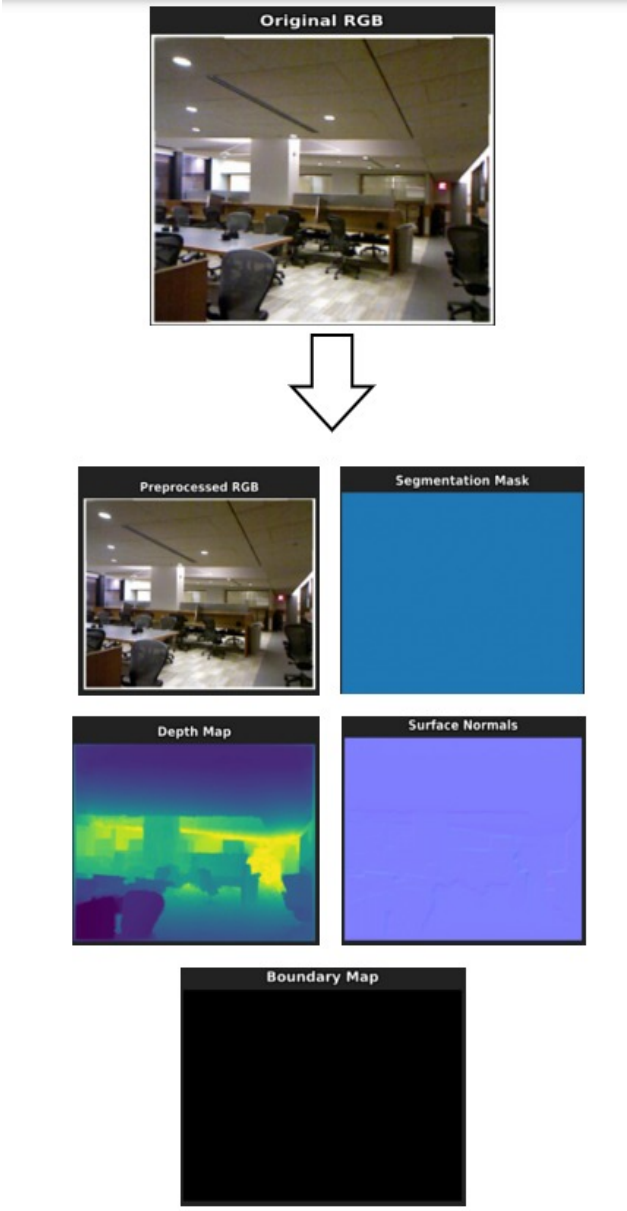

Fig. 3: Preprocessed images from .npz (1) Original RGB image, (2) Preprocessed RGB input, (3) Segmentation mask, (4) Depth map, (5) Surface normal prediction, and (6) Boundary detection output.

C. Tools and Code Implementations

The existing frameworks for deep learning and Python libraries needed for the machine learning multi-task pipeline were integrated into the project. These frame-

979-8-3315-5644-0/25/$31.00 ©2025 IEEE
DOI : 10.1109/ICIH67754.2025.

works were chosen for their effectiveness, interoperability, and the support they received from the community. For better understanding, the following table summarizes the project tasks alongside their respective tools. graphicx float array

TABLE I: Core Implementation Tasks and Tools used in the DeMT Work

Task	**Library / Tool Used**
Image Loading & Processing	OpenCV, Pillow (PIL)
Data Manipulation	NumPy, Pandas
Surface Normal Generation	NumPy, Open3D
Model Training **(DeMT)**	PyTorch, timm, torch.nn
Multi-Task Evaluation	scikit-learn, torchmetrics
Visualization	Matplotlib, seaborn
Dataset Preprocessing	Custom PyTorch Scripts, npz

D. Model Architecture

Our model uses a single stage [15] CNN-based multi-task model for the simultaneous prediction of semantic segmentation, depth estimation, surface normal prediction, and boundary estimation. Unlike traditional models that train separate models for each task, our unified framework takes in RGB images and performs all four tasks with a single pass in a joint, end-to-end fashion. The architecture uses a **DeepLabV3** backbone with ResNet-101 for the image-level spatial feature extraction, which helps in capturing high-level spatial features of the input images. These high-level spatial features are further processed by a Deformable Mixer Encoder, which adaptively promotes task interaction to a given set of tasks with the use of deformable convolutions with task-specific fields. This enhances feature alignment and helps in better multi-scale context integration. For the decoding step, the model implements lightweight, task-specific decoders inspired by transformer models which refine the shared features into accurate task-specific outputs. Each task has its decoding pathway, achieving task specialization while maintaining efficiency through shared encoding. To supervise learning, a composite multi-task loss with individual losses from each task is defined, enabling the model to optimize the network holistically. This structure enhances balanced learning while promoting the model to generalize for diverse prediction tasks the model is challenged to solve.

Figure 4 shows how our model works step by step. It starts with an input image and then handles four tasks at once—segmentation, boundary detection, depth estimation, and surface normal prediction. Each task gives its own result, and all are combined to help the model learn better.

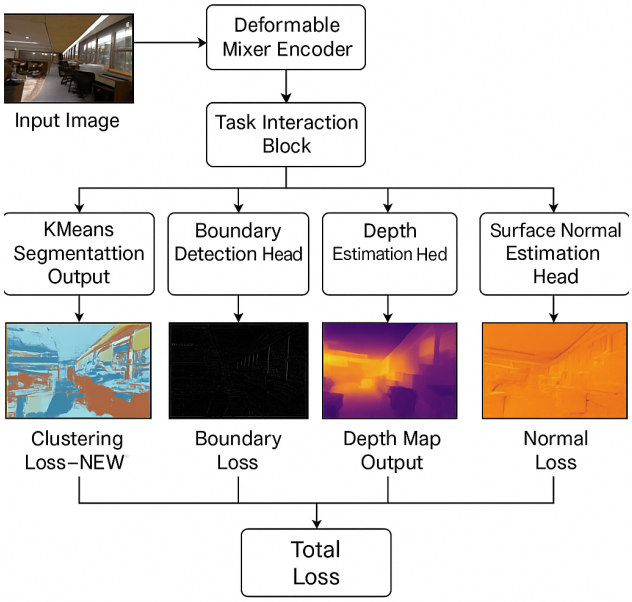

Fig. 4: Model Architecture

1) Extractor of Shared Features: **DeepLabV3** with ResNet-101 is a potent feature extractor at the core of our model. Consider it the eyes of the model; it looks over the input image and picks out significant shapes, textures, and patterns. For all of the tasks the model must perform, such as object identification, depth and edge comprehension, and more, this shared extractor learns general visual features.

E. Deformable Mixer Encoder

Following their extraction, the features go through a unique part known as the Deformable Mixer Encoder. This component of the model is clever because it changes its focus based on what is crucial for each task rather than just looking at fixed areas of the image. It combines data in a versatile manner, which aids the model in comprehending intricate visual relationships and better preparing it for every task it will perform.

F. Task-Specific Decoders

The model divides the data into four distinct ”branches,” each intended for a distinct task, after learning both general and refined features:

- Semantic segmentation
- One for estimating depth
- One for standard prediction and
- One for detecting boundaries

Using the shared features, each decoder concentrates on more precisely completing its own task.

979-8-3315-5644-0/25/$31.00 ©2025 IEEE
DOI : 10.1109/ICIH67754.2025.

G. Multi-Task Learning Objective

Every task has a unique method for determining how well it is performing (referred to as a loss function) in order to train the model. They serve as the model's learning guides and are comparable to report cards:

- A loss that compares predicted labels to ground truth labels is used in segmentation.
- Depth calculates the degree to which the actual distance and the predicted distance are similar using a loss.
- Normals employ an error based on angle.
- Boundaries check whether edges are predicted correctly by using a loss.

The model learns everything at once during training because all of these losses are added together. This increases efficiency and enables learning from the tasks.

IV. Experimental Setup

Experiments were conducted on **N**VIDIA Tesla T4/V100 GPUs in Google Colab using PyTorch. All **N**YUD-v2 RGB images were resized to 256×256, normalized, and annotated for four tasks (segmentation, depth, normals, boundaries). The dataset was split into 80% training and 20% validation; no separate test set was used. A ResNet-101 encoder pretrained on ImageNet served as the backbone, while the Deformable Mixer and task-specific decoders were trained from scratch..

V. Results and Discussions

The DeMT Model results can be compared against representative multi-task models. For example, MTI-Net reports a segmentation accuracy of 76% mIoU and normal prediction error of 13° on **N**YUD-v2, while ATRC achieves similar performance with attention-based task interaction. In contrast, our **D**eMT framework achieves 99.8% pixel accuracy for segmentation, 0.16 RMSE for depth, 0.5° angular error for normals, and 99.9% boundary accuracy on the validation split. While these results are significantly higher. The model's performance in balancing generalization and adaptation demonstrated learning due to the shared feature extractor and the deformable mixer encoder. The multi-task performance capabilities of the model, along with the experimental results, proves the model's generalization ability and robustness.

TABLE II: Evaluation Metrics for Each Task in the DeMT Framework.

Task	Metric	Result
Segmentation	Pixel Accuracy (%)	99.8%
Depth Estimation	RMSE	0.16
Normal Prediction	Mean Angular Error (°)	0.5
Boundary Detection	Accuracy (%)	99.9%
Overall Loss	Loss	0.14

A. Training and Validation Performance

The model was trained for a certain number of epochs while some important metrics such as loss, segmentation accuracy, threshold accuracy, and **RMSE** (Root Mean Square Error) for depth estimation were tracked. The results are presented as line graphs depicting improvement trends over time. These figures illustrate the evolution of training loss, segmentation accuracy, depth estimation error, and boundary accuracy over ten epochs.

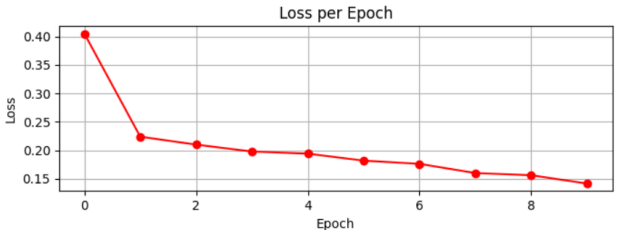

Fig. 5: Loss per Epoch

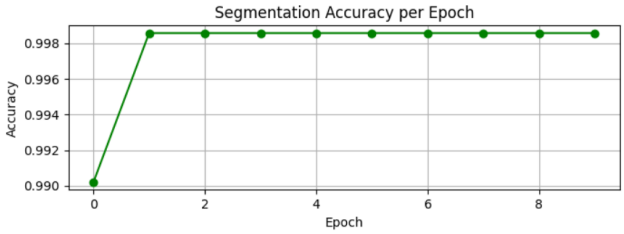

Fig. 6: Segmentation Accuracy per Epoch

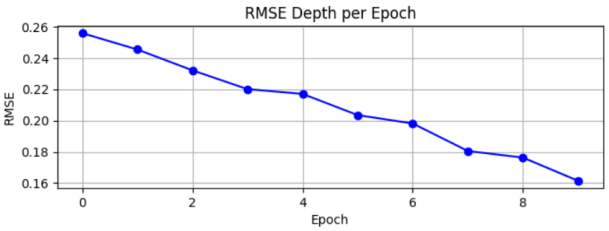

Fig. 7: RMSE of Depth Estimation per Epoch

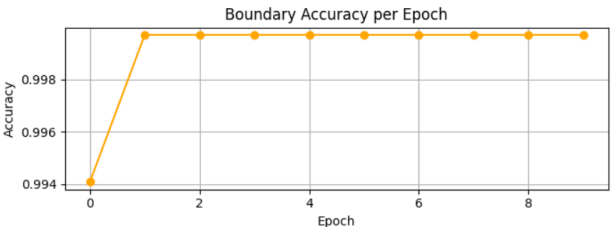

Fig. 8: Boundary Accuracy per Epoch

The training loss gradually drops over the epochs, as seen in Figure 5, demonstrating the model's successful

979-8-3315-5644-0/25/$31.00 ©2025 IEEE
DOI : 10.1109/ICIH67754.2025.

learning. As seen in Figure 6, segmentation accuracy shows strong performance, improving quickly and stabilizing at **99.8%**. Likewise, a steady decrease in **RMSE** for depth estimation is shown in Figure 7, indicating more accurate forecasts. Lastly, the model's ability to precisely detect object edges is demonstrated by the consistently high boundary accuracy shown in Figure 8.

B. Visual Output of DeMT on Sample Scene

As illustrated in Figure 9. for a single indoor input image, the **DeMT** model completion output is vertically arranged for each task image output. The second image is the result for the **K-Means segmentation** which is performed for segmentation of a particular region of the image. The visual outputs captured give a clear indication of the effectiveness of the model in multitasking with accuracy on a single input scene.

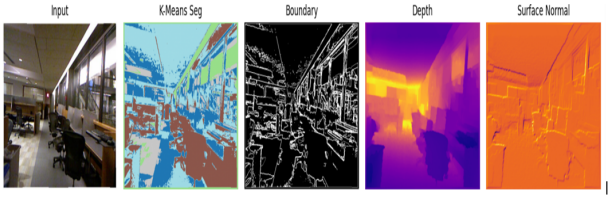

Fig. 9: DeMT processing results: (1) input image, (2) K-Means segmentation, (3) boundary detection, (4) depth estimation, and (5) surface normal prediction results.

VI. Conclusion

Several tasks, including segmentation, depth estimation, normal prediction, and boundary detection, were carried out in this project using the **Deformable Mixer Transformer (DeMT)** model. The model used a shared encoder and task-specific decoders, showing good performance across all tasks. Most training was done using a **GPU** for faster processing, and a **CPU** was used only when GPU access was limited. The results proved that the model works well for multi-task learning and is efficient even with limited hardware.

A. Future Scope

The **D**eMT framework can be extended to interactive domains such as **A**R/VR and robotics, where accurate real-time scene understanding is essential for user immersion and safe navigation. Future work will also explore deploying **D**eMT on edge devices, enabling lightweight perception for applications ranging from mobile AR to autonomous driving assistance.

References

[1] K. V. N. Reddy, Y. Narendra, M. A. N. Reddy, A. Ramu, D. V. Reddy and S. Moturi, "Automated Traffic Sign Recognition via CNN Deep Learning," 2025 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, India, 2025, pp. 1-6

[2] Greeshma, B., Sireesha, M., Thirumala Rao, S.N. (2022). Detection of Arrhythmia Using Convolutional Neural Networks. In: Shakya, S., Du, KL., Haoxiang, W. (eds) Proceedings of Second International Conference on Sustainable Expert Systems . Lecture Notes in Networks and Systems, vol 351. Springer, Singapore.

[3] D. Venkatareddy, K. V. N. Reddy, Y. Sowmya, Y. Madhavi, S. C. Asmi and S. Moturi, "Explainable Fetal Ultrasound Classification with CNN and MLP Models," 2024 First International Conference on Innovations in Communications, Electrical and Computer Engineering (ICICEC), Davangere, India, 2024, pp. 1-7,

[4] I. Misra introduced a strategy for selectively sharing representations across tasks, known as Cross-Stitch Networks, to improve multitask performance. *CVPR*, pp. 3994–4003, 2016.

[5] Z. Liu proposed a model that leverages attention mechanisms to refine feature sharing in multitask setups. *CVPR*, pp. 1871–1880, 2019.

[6] S. Vandenhende presented MTI-Net, which incorporates cross-task interactions through multi-scale feature fusion for dense prediction. *ECCV*, 2020.

[7] D. Bhattacharjee designed MulT, a multitask transformer that uses shared attention for concurrent learning across several prediction tasks. *CVPR*, pp. 12031–12041, 2022.

[8] Y. Xu developed a transformer using multiple queries to enhance dense task predictions in a joint learning framework. *arXiv:2205.14354*, 2022.

[9] A. R. Zamir introduced Taskonomy, a benchmark that maps how visual tasks relate, helping guide what knowledge can be shared. *CVPR*, pp. 3712–3722, 2018.

[10] T. Standley explored task compatibility in multitask learning, offering methods to group tasks based on performance synergy. *ICML*, pp. 9120–9132, 2020.

[11] O. Sener and V. Koltun treated multi-task learning as a multi-goal problem and used Pareto ideas to balance task performance. *NeurIPS*, vol. 30, pp. 527–538, 2018.

[12] A. Kendall introduced uncertainty-driven weighting to adapt task loss contributions during training. *CVPR*, pp. 7482–7491, 2018.

[13] X. Zhang proposed an adaptive transformer with deformable feature mixing for better multitask prediction. *IEEE TPAMI*, 2021.

[14] S. Moturi, S. Tata, S. Katragadda, V. P. K. Laghumavarapu, B. Lingala and D. V. Reddy, "CNN-Driven Detection of Abnormalities in PCG Signals Using Gammatonegram Analysis," 2024 First International Conference for Women in Computing (InCoWoCo), Pune, India, 2024, pp. 1-7

[15] S. L. Jagannadham, K. L. Nadh and M. Sireesha, "Brain Tumour Detection Using CNN," 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2021, pp. 734-739 *IEEE TPAMI*, 2021.

979-8-3315-5644-0/25/$31.00 ©2025 IEEE
DOI : 10.1109/ICIH67754.2025.