IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 November 2022, accepted 25 November 2022, date of publication 30 November 2022,
date of current version 6 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3225746

RESEARCH ARTICLE

MBMT-Net: A Multi-Task Learning Based Convolutional Neural Network Architecture for Dense Prediction Tasks

GEORGE CIUBOTARIU[1,2] **AND GABRIELA CZIBULA**[1]
[1]Department of Computer Science, Babeş-Bolyai University, 400084 Cluj-Napoca, Romania
[2]Robert Bosch SRL, 400158 Cluj-Napoca, Romania

Corresponding author: George Ciubotariu (george.ciubotariu@stud.ubbcluj.ro)

The research leading to these results has received funding from the NO Grants 2014-2021, under Project contract no. 26/2020.

ABSTRACT Recently proposed improvements in the field of Computer Vision refer to enhancing the feature processing capabilities of Single-Task Convolutional Neural Networks. A typical Single-Task network consists of a backbone and a head, where the feature extractor is usually optimised using the gradient provided by the head. Inevitably, the backbone specialises for the given task. This sort of approach does not scale well for learning multiple tasks at once while having the same input. As a response, there is an increasing interest in Multi-Task formulations. Since most Multi-Task architectures employ a single shared backbone, when gradients from different tasks are propagated back to it, it can result in its oversaturation. Thus, this problem may be solved using Multi-Backbone feature extractors. Hence, as a strategy proposed to compensate for these shortcomings, we introduce *MBMT-Net*, a Multi-Backbone-Multi-Task-Network architecture based on a development strategy that infuses backbones with more diverse and specialised processing capabilities. *MBMT-Net* consists of parallel pre-trained backbones whose outputs are concatenated and offered to the Multi-Task heads that shall benefit from richer and more diverse features with decreased number of network parameters when compared to traditional Multi-Task architectures. Our strategy is architecture independent, and it can be applied to different types of backbones and parsing heads, which greatly extends the domain of configurable features, finally enhancing existing Single- and Multi-Task model building strategies and outperforming them when using the Multi-Backbone design. As a result, while having a deficit of 12.16M parameters, *MBMT-Net* reaches state-of-the-art performances, and surpasses the previously best semantic segmentation Multi-Task model in terms of Mean Intersection over Union when evaluated on NYUv2 data set.

INDEX TERMS Computer vision, convolutional neural network, depth estimation, multi-backbone, multi-task, semantic segmentation, surface normal prediction.

I. INTRODUCTION

The growing interest in Computer Vision (CV) [1] is developing upon the rapid progress of computational means for very complex learning models. At the same time, the research motivation lies on the need of process automatisation in several domains in which Machine Learning (ML) is applied. Furthermore, the present offers us many open questions, and unsolved problems, that we can tackle using ML models, especially the ones that perform multiple tasks at once by employing several means to extract essential features. Consequently, there is a strong desire to improve the current state of knowledge and outperform humans' decision making process in difficult tasks.

Recent advancements in the field of CV refer to enhancing the feature processing capabilities of Single-Task (ST)

The associate editor coordinating the review of this manuscript and approving it for publication was Zhipeng Cai.

Convolutional Neural Networks (CNN). Consequently, novel approaches were proposed, as the interest in real-life ML applications grew. For addressing the problem of finding the most suitable designs, researchers developed models based on single-task architectures. Therefore, such networks achieved state-of-the-art results in solving dense prediction tasks such as our tasks of choice, namely Semantic Segmentation (SS), Depth Estimation (DE), and Surface Normal Prediction (SNP). However, the problem formulations may be suboptimal in regards of the learning context the models are put in. A typical ST network consists of a backbone and a head, where the feature extractor is usually optimised using the gradient provided by the head. Inevitably, the backbone specialises for the given task. This sort of approach does not scale well for learning multiple tasks at once while having the same input. As a response, there is an increasing interest in Multi-Task (MT) formulations.

On the one hand, since the CNN architecture search is an intensively researched topic, many papers have focused on improving and designing new building blocks [2], [3]. A reason for that is the powerful hardware components which we are provided with. They facilitate training deep networks [4], and properly developing CNN architectures. Hence, such breakthroughs lead towards state-of-the-art results on most popular data sets. On the other hand, we believe that while the lower-level layers improve various structural elements of a model, on a higher level there may still be room for increasing performance. That may consist of macro model fine-tuning, such as combining several building techniques to boost the benefits, or to cover each others' shortcomings. Consequently, additional training time supervision [5], auxiliary losses [6], specialised learning mechanisms [7], and richer learning contexts could be implemented in order to maximise the gains of a network, at the expense of building more complex systems. Among the mentioned techniques, the latter may prove to be effective especially in MT contexts, and it involves the least development resources, by combining several model parts together. Nonetheless, recent works on MT learning have to add extra parameters to the CNN [8], so that their architectures could specialise on several tasks that, when jointly trained, help each other perform better when compared to the standalone counterparts. Considering that, without employing supplementary mechanisms, or changing the ST models' default components, MT networks rethink the learning contexts so that they could benefit from jointly training several related tasks at once.

Multi-task networks have gained popularity in the recent years, once the demand of complex image processing systems took off. Therefore, they differentiated themselves from traditional encoder-decoder architectures by including more parsing heads responsible with solving particular tasks that are related to each other. The features learned in this common context will be successfully used for all the tasks in order to increase the performance. In doing so, the core concepts underlying dense prediction tasks are deepened in the embedded feature representation, which results in a higher level of robustness when observing certain frequent patterns. As a result, the multi-task learning context offers various verification means, leading to more confident predictions.

Hence, the usefulness of MT-CNNs becomes obvious, due to the sheer collaboration between the parsing heads and the shared backbone, which resembles the human way of learning related tasks in parallel. However, this type of networks could only learn so much out of these contexts because of the *oversaturation* problem. As shown in our experiments from Section VI, this is a real issue that considerably affects model performance. This phenomenon is a consequence of insufficiently complex ML models, that are progressively requested to perform more than they are able to manage. Without an appropriate backbone size, the training progress of the models would stall at some point in time. That happens because the heads would not be able to benefit from the condensed representations, as underfit models yield insufficiently processed features. This problem also appears in a different scenario, when the backbone is deep enough, but the too shallow heads cannot particularise the general feature maps. Regarding this statement, we have performed extensive experiments in this direction for non-dense prediction tasks learning, and we have not been able to learn anything unless we used the thoroughly processed multi-scale features.

The ideas that motivate us to research this direction are the following. Firstly, learning multiple tasks at once ensures a better image context understanding. Secondly, for keeping under control the model's number of parameters, the layer dimensions should be reduced, which results in oversaturation. In the third place, what we consider that a multi-task architecture lacks is collaboration between multiple backbones. When increasing the number of parsing heads, such architectures hardly scale well, and oversaturation occurs. A potential solution is to add backbones to capture more useful features. Besides the saturation problem, every encoder must learn a different task, for offering the rest of the network more contextual information, without losing significant details. In doing so, the model would have additional verification possibilities regarding the assumptions each standalone network must make, which ultimately aids the joint training schedule to build knowledge on correct and consistent foundations.

In this paper we are proposing a solution for building MT-CNNs, and achieving appropriate fitting by developing an architecture-invariant strategy. It consists of replacing the single shared backbone with multiple shallower, specialised backbones. In the proposed multi-backbone (MB) and multi-task architecture, named *MBMT-Net* (**M**ulti-**B**ackbone and **M**ulti-**T**ask **Net**work), the pre-training is done in ST networks, then each backbone is put together in the MB-MT-CNN, and optionally frozen. When frozen, our training schedule dramatically reduces the GPU memory consumption. This allows training the network on lower-end hardware without sacrificing the final performance, but increasing it instead. We conceptually prove the effectiveness of our strategy by implementing *MBMT-Net*, a three

backbones and three heads CNN that we experiment with on the pre-processed NYUv2 data set [9] offered by [10]. Yudong et al. [11] are also using multiple backbones, as in our proposal. However, our approach differs from the previous one [11], as we aim to allow any combination of processing units in the learning context, not just identical structures. Moreover, our goal is to preserve the number of the multi-backbones parameters similar to the one of a traditionally structured MT model, without trading performance off. Additionally, what differentiates our *MBMT-Net* approach from other methods in MT learning is that the usual single shared backbone is replaced with multiple specialised, independent ones.

As a result, our method describes an MT-CNN design alternative, having a broader range of applications, and extension opportunities for further research in the Distributed Artificial Intelligence (DAI) field. To the best of our knowledge, the *MBMT-Net* architecture is new in the CV literature.

In summary, the paper is addressing the following research questions:

RQ1 How to enhance the performance of dense prediction tasks by using the multi-task learning paradigm? In this respect, we are introducing the *MBMT-Net* model.

RQ2 To what extent does the *MBMT-Net* improve the performance of current state-of-the-art approaches in dense prediction tasks?

RQ3 Is the performance improvement achieved by *MBMT-Net* with respect to existing solutions statistically significant?

The rest of the paper is organised as follows. Secondly, Section III briefly discusses the architecture designs of CNNs. Thirdly, Section IV puts forward the methodology followed in the experiments, and presents the changes specific to the *MBMT-Net* architecture. The data set and the experimental setup employed for the evaluation of the *MBMT-Net* model are presented in Section V. Then Section VI presents the experimental findings, and underlines the improvements brought in this study. Eventually, Section VII aims to summarize the conclusions of our study, to pinpoint the answers to the introduced research questions, and to identify directions for future improvements.

II. TASKS IN FOCUS

This section outlines our choice regarding the tasks we have selected to solve. In doing so, we summarily explain how each of them is generally supposed to be approached.

A. SEMANTIC SEGMENTATION

Semantic Segmentation (SS) is one of the most popular in the research community. It requires a model to process an input image, and assign each pixel the semantic class it belongs to. Mainly, it is used to classify regions, and understand the scene, so that other systems could make decisions based on the findings [12], [13].

B. DEPTH ESTIMATION

Another difficult dense prediction task is Depth Estimation (DE) [14], which is recognised in the literature as being ill-posed, since depth cannot be fully recovered from a single image without environment-specific assumptions. Compared to the previous one, this is often formulated as a regression problem, as the depth values are continuous, and they belong to a pre-determined interval [15], [16].

C. SURFACE NORMAL PREDICTION

The third task is Surface Normal Prediction (SNP), which is also a regression task, and it supposes that each point in the raw image is assigned an RGB value corresponding to angles of the surface normal vector in the 3D space. This is useful in determining the shape of the objects, which increases the predictions consistency when having priors about the scene's content. However, it requires finer grained context understanding, because many features of the objects present in the scene are occluded. Nonetheless, depending on the nature of the data set, when reconstructing the initial image resolution, models may require more than basic interpolations to preserve the information when solving critical importance tasks [17], [18].

III. RELATED WORK

This section introduces, and describes the tasks we are going to focus on in the experiments. Furthermore, in order to better understand the architectural decisions we make in the *MBMT-Net* development, we will present several state-of-the-art architectures that led us to the final shape of our model.

A. SINGLE-TASK NETWORKS

CNNs have been used many times with great success in solving CV dense prediction tasks. Many papers refer to them as the ''workhorses'' of neural networks. Therefore, we focused our attention on working with such models in this paper.

A robust architecture we considered is EfficientPS [19], that employs a two-way Feature Pyramidal Network (FPN) [20], which uses the multi-scale features more effectively. The authors' contribution can be observed in the novel SS parsing head, which consists of dense prediction cells [21], and residual pyramids. Thanks to the model's ability to capture fine features, long-range contextual features, and because it correlates the distinctly captured features, it improves object boundary refinement. Hence, this CNN model has great potential for being used in multiple dense prediction tasks, since the elaborated context understanding techniques would provide reliable, and consistent feature processing, regardless of the CV task.

B. MULTI-BACKBONE NETWORKS

Motivated by the idea of modeling extra dependencies between higher-level features, a couple of recent articles use several encoders in their multi-backbone architectures. Their advantage is that of being able to extract richer and

finer-grained features from data, through the thorough processing of the input, which results in higher overall model performance.

The first paper to introduce a technique of assembling several identical backbones is CBNet [11], therefore significantly increasing the number of parameters. Their goal is to include higher-level features into succeeding backbones, to boost its context understanding, and final receptive field. This is achieved by using the help of skip connections that are similar to transposed convolutions, as the compound backbone structure resembles an unraveled, limited cycle Recurrent Neural Network (RNN) [22].

The best performing classification model that uses compound backbones is the improvement of the former one, namely [5]. Its architecture is implemented using an optimised feature sharing scheme across encoders via skip connections at training time. Instead of the former model, at evaluation time, the additional backbones are removed for the latter architecture variant. This strategy is constrained to use identical backbones, as the feature maps sharing cannot be performed otherwise.

C. MULTI-TASK NETWORKS

In contrast to the aforementioned topic, multi-task [23] networks use multiple parsing heads so that each of them focuses on simultaneously learning, and performing separated tasks.

The feature extractor part of the model can look differently, but most of the times there is a single shared backbone. It provides all the parsing heads with condensed feature representations they can further particularise to solve their specific task.

Nevertheless, there are two multi-task building strategies, which refer to the way they share parameters [24], [25], [26]. Firstly, *Hard Parameter Sharing* refers to using a single backbone (SB) which is shared by all the parsing heads. This is particularly useful for training multiple tasks together, as it acts as a regulariser, and the compressed representation the model learns should better generalise for the specific tasks. Secondly, *Soft Parameter Sharing* does not have any fully shared component, but it rather has an own structure for each task, and during learning, it penalises the difference between the models' parameters. The constrained layers are therefore encouraged to have similar weights values for related parameters. However, this significantly increases the parameter usage and slows down the training. Although less sophisticated, the hard parameter sharing is often sufficient for establishing a favourable learning environment, and enhance the robust feature extraction abilities of a model.

Consequently, a recent multi-task architecture (MTAN) that sets a new state-of-the-art (SOTA) performance for multiple dense prediction tasks has been introduced by Liu et al. [10]. It is built using a hard parameter sharing strategy, and it consistently benefits from several novelty elements, such as the soft-attention mechanism, overall multi-task loss, and dynamic task convergence speed adjustment. We also found the latter improvement very useful in our MT learning context, and implemented it to balance task learning, so all the individual parsing heads have the chance to learn together, and avoid early and asynchronous convergence rates, which would determine easier tasks to overfit.

Having presented the *multi-backbone*, and *multi-task* approaches, we can see the importance of these improvements, and the potential of combining structural pieces of the both. As a result, these two model building strategies bring their advantages to the same learning context. However, there are multiple difficulties in building such models, since putting together the processed feature maps may require extra caution.

The development of a robust multi-task model is difficult, and represents a challenging research topic. A robust model must be capable of processing large volumes of data, not oversaturate, and efficiently manage its resources.

In the literature, there are diverse approaches for solving CV problems using CNNs. During the last years, MT learning has benefited from great attention in the research community, as it achieved results superior to the single-task models, thanks to the joint learning context. However, no model in the literature is implemented using an architecture similar to the one we proposed for solving CV tasks.

IV. METHODOLOGY

In this section we are introducing *MBMT-Net* architecture. It is a CNN consisting of three backbones and three heads. Our goal is to offer a solution for building MT-CNNs, by developing an architecture-invariant strategy. What differentiates our approach in multi-task learning is that the usual single shared backbone is replaced with multiple specialised, independent backbones. Moreover, we chose a two-step training schedule that allows us to capture more useful features. Firstly, we pre-train each backbone in ST networks, then each of them is put together in the MB-MT-CNN, and optionally frozen. When frozen, our training schedule dramatically reduces the GPU memory consumption. This allows training the network on lower-end hardware without sacrificing the final performance, but increasing it instead.

An extensive study is necessary for multi-backbone architecture usage in multi-task learning context. However, the overall performance in this scenario is proportional to the number of parameters. Nonetheless, by using multiple encoders with reduced depth, we support feature diversity in the condensed representation. Therefore, this offers the task-specific parsing heads a more robust concatenated feature map that shall gather visual cues from all the separated representations, which may cover each others' insufficiently informative extracted patterns. Additionally, we focused on designing the multi-backbone blueprint in a low-coupled fashion. There is only one constraint regarding the final shape of the independent encoders, so that the resulting tensors can be merged together. Therefore, the architecture blueprint has a great extension potential, as any backbone can be used, as long as it provides output feature maps compatible with the others.

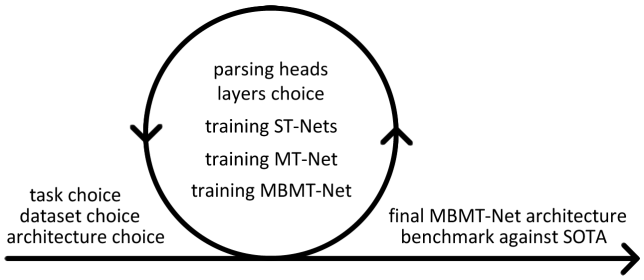

FIGURE 1. Development methodology of *MBMT-Net*.

Therefore, we are introducing in the following the methodology for developing the *MBMT-Net* architecture, and setting up the training, and evaluation.

Figure 1 schematically describes the process we have followed, in order to achieve the final *MBMT-Net* results. In the beginning, we decided on performing several dense prediction tasks at once. Then, we have have chosen the EfficientPS [19] architecture. It is easy to extend and it suits the performance requirements to compete with MTAN [10]. Consequently, we decided to adapt EfficientPS to a multi-task learning context, and to perform the same tasks as MTAN on NYUv2 [9].

Afterwards, we have continuously improved our architecture, and fine-tuned the hyper-parameters to achieve the best results. More on the training methodology will later be explained in Section IV-B. Hence, by performing a grid search on the possible values of several parameters, and comparing models' performances from one iteration to another, we obtained the optimal setup for our *MBMT-Net* architecture. Additionally, we have changed layers, activation functions, and upsampling means. Eventually, we obtained the final *MBMT-Net* architecture depicted in Figure 2. Finally, we performed several more trainings of the models, and compared the final results to MTAN [10], which is the current SOTA multi-task architecture.

A. ARCHITECTURE DESIGN

With the aim to develop a novel approach for solving CV tasks, our MB-MT model designing strategy on CNN architectures will be further introduced. Starting from the intuition that a model would perform better when provided with the means of regarding the same scene from different perspectives, we believe that it would use the insight to validate extracted patterns, yielding robust results. Additionally, the major benefit for such a model is that it would build up what we call a certain degree of intuition, and common sense of the *MBMT-Net* network, that us humans also use unconsciously. Hence, having learned multiple recurring patterns of the different CV tasks, the *MBMT-Net* model we propose would be able to more confidently solve difficult problems.

Considering that we are working with multiple feature maps, the output values of the multi-backbone have to be merged together. This raises the question of which feature maps merging technique is the most suitable for our problem. Hence, we are going to approach three of the most common ones, namely concatenation (CAT), addition (ADD), and multiplication (MUL). We will later refer to these operations in Table 1, in the MT variants and architecture abbreviations, by adding -CAT, -ADD, or -MUL at the end of an architecture abbreviation. For instance, *MBMT-Net*-CAT refers to the *MBMT-Net* architecture that employs the concatenation operation for feature maps merging.

Our work stands as a proof of concept that should emphasize the benefit of multiple backbones in a multi-task learning context. Table 1 summarizes the usage, and components of the models that use EfficientPS variants. We note that the notation B1 used in the last column from the table means a single backbone, while the notation MBn (with $n > 1$) refers to n backbones (i.e., MB3 refers to three backbones). The MT architectures use the Dynamic Weight Average (DWA) weighting scheme presented in [10]. By observing the rate of change of the loss for each task, it adjusts the task weighting over time. In further experiments we will refer to the models using their corresponding abbreviation.

The research we are performing tackles more complex scenarios than the automated neural architecture search (such as AutoML [27], [28]) could handle. Not only that we perform hyper-parameter tuning, and training schedule tuning in our experiments, but our contribution also refers to proposing a generalised model building strategy. Instead of searching for the best model to solve our problem, we aim to improve the concept of multi-task learning via our approach.

For implementing our models and running the experiments, we have used the PyTorch [29] framework. The input resolution of the models is 288 $\times$ 384 pixels, as Liu et al. [10] offered us the NYUv2 images directly converted into a convenient format, so that we could create a fair benchmark. All the EfficientNet [30] backbones variants have been downloaded from the links provided by the authors and pre-trained on ImageNet [31]. EfficientPS [19] model has been used as a starting point for our analysis.

The two-way FPN proposed in EfficientPS [19] is depicted in Figure 2 through the blue and purple branches. While the former branch follows the conventional FPN aggregation scheme, namely from right to left, the latter downsamples the higher resolution features to the next lower resolution in the opposite direction.

Some of the more complex *MBMT-Net* building blocks are presented in Figure 3. Firstly, the modified Dense Prediction Cells (DPC) module used by us employs Leaky ReLUs as activation functions instead of their original ReLUs. It is a more effective variant of the Atrous Spatial Pyramid Pooling (ASPP) module. Secondly, the Large Scale Feature Extractor (LSFE) module is employed in order to efficiently capture fine features. Eventually, the Mismatch Correction Module (MC) aims to mitigate the mismatch between large and small-scale features when aggregating the extracted features.

Regarding the ST designs, we have removed the *Instance Segmentation Head*, as we reduced the complexity from

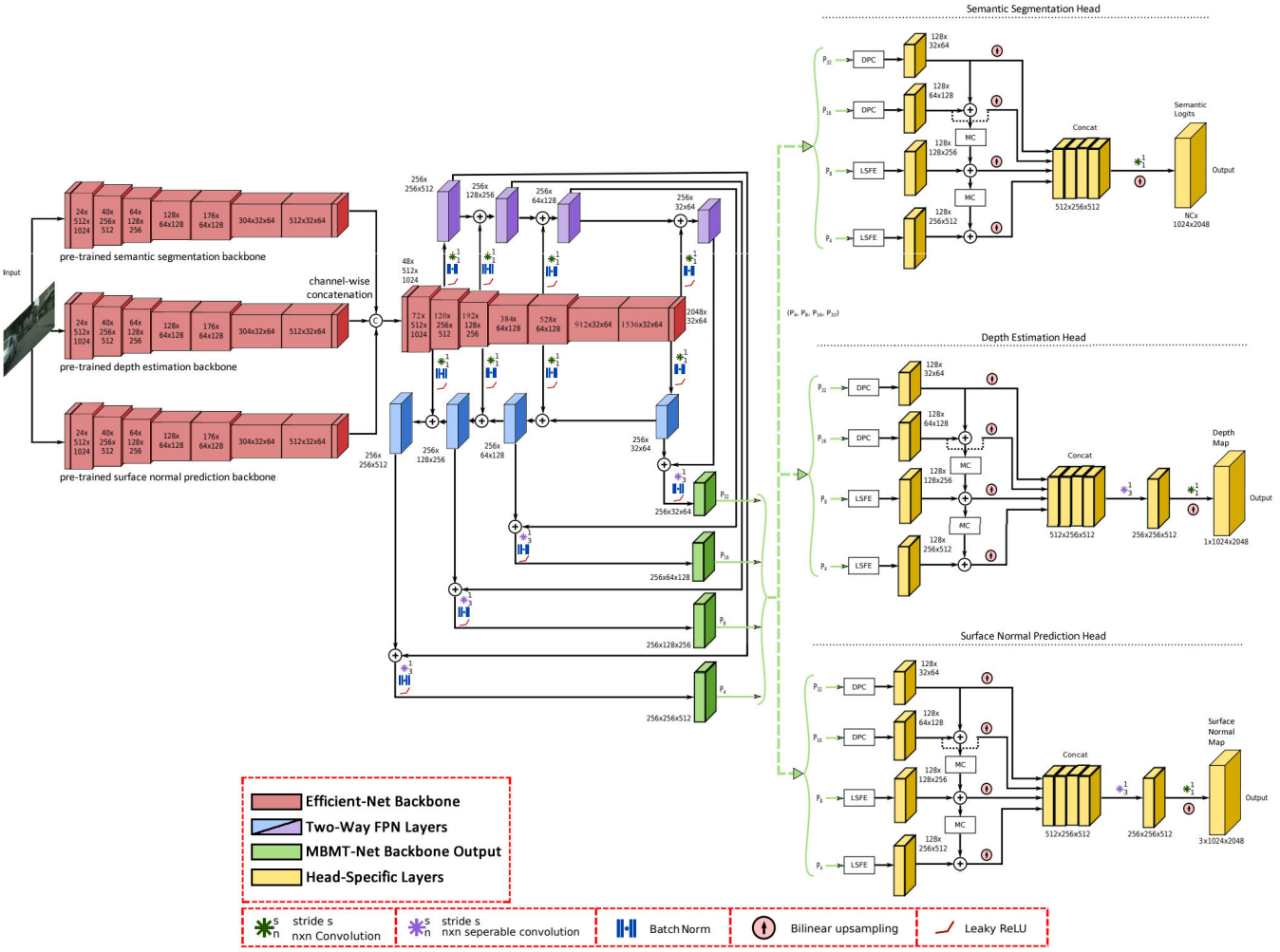

FIGURE 2. **Architectural overview of *MBMT-Net*. The figure is based on the one presented in.**

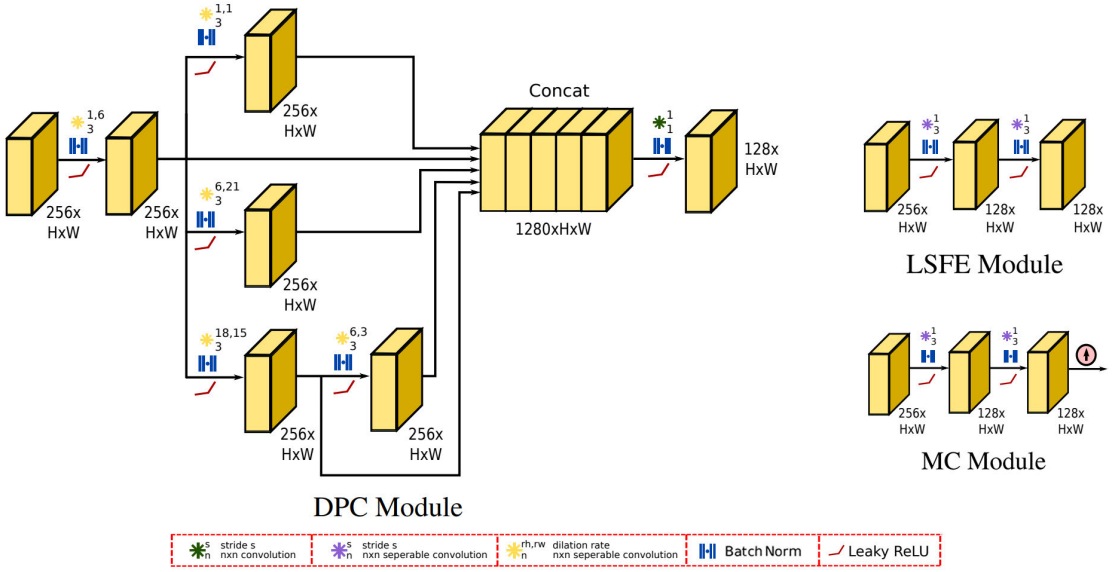

FIGURE 3. **Topologies of various head-specific architectural components of *MBMT-Net*.**

TABLE 1. The architectures used in our experiments.

Context	Backbone	Architecture	Tasks	Features Merging	Abbreviation	MT Variant
Single-Task	1xEfficientNet-b2	EfficientPS	SS	-	ST-SS-Net	-
			DE		ST-DE-Net	
			SNP		ST-SNP-Net	
Multi-Task	1xEfficientNet-b5	MT-EfficientPS DWA	SS, DE, SNP	CAT	MT-Net	B1
Multi-Backbone Multi-Task	2xEfficientNet-b2	MT-EfficientPS DWA	SS, DE	CAT	*MBMT-Net* no SNP	MB2-xSNP
			SS, SNP		*MBMT-Net* no DE	MB2-xDE
			DE, SNP		*MBMT-Net* no SS	MB2-xSS
Multi-Backbone Multi-Task	3xEfficientNet-b2	MT-EfficientPS DWA	SS, DE, SNP	ADD	*MBMT-Net*-ADD	MB3-ADD
			SS, DE, SNP	MUL	*MBMT-Net*-MUL	MB3-MUL
			SS, DE, SNP	CAT	*MBMT-Net*-CAT / *MBMT-Net*	MB3

TABLE 2. Summary of the parsing head specific convolutions.

	stride	padding	kernel	activation
SS Head	1	0	1×1	softmax
DE Head	1	1	3×3	identity
	1	0	1×1	identity
SNP Head	1	1	3×3	identity
	1	0	1×1	identity

panoptic segmentation to semantic segmentation by using only the EfficientNet-b2 encoder [30], and the *Semantic Segmentation Head* from the original network. The output are the semantic logits in the case of SS. For the other two tasks (i.e., DE and SNP), we changed the last two convolutions, and activation functions to fit the kind of problem we deal with.

The differences between the three can be clearly visualized in Figure 2, as it presents the fully convolutional *MBMT-Net* having 3 backbones, and 3 parsing heads for the SS, DE, and SNP tasks. Thus, the architectural differences are the following. While SS head employs a 1×1.2D convolution layers (Conv2D) which compresses 512 to *nb_class* channels, activated by a softmax function, the DE and SNP heads use two Conv2D layers, namely a 3×3, reducing 512 to 256 channels, and a 1×1 similar to the SS one, but with no activation. We also note that the number of classes (*nb_class*) varies between the tasks, as DE outputs single-channel distances, and SNP outputs the 3-channels angle values encoded into an RGB map.

Table 2 briefly presents the architectural design of the three parsing heads. We must mention that the output of the final convolutions is upsampled 4 times via bilinear interpolation. Additionally, after upsampling, the SNP head's output is divided by its Frobenius norm.

Then, for the SB-MT experiments, we decided to use an EfficientNet-b5 backbone. The two-way FPN output feature maps are offered to each of the SS, DE, and SNP heads which process them and output their predicted masks.

Eventually, in the MB-MT experiments, a weaker backbone variant of the EfficientNet, namely the b2 was used so that the total number of encoder parameters would add up to roughly the same as the one for the single EfficientNet-b5. Each of the EfficientNet-b2-s are pre-trained in an ST context until the loss flattens, then they are put together in the *MBMT-Net*. Their outputs must be channel-wise concatenated, and fed into the two-way FPN module. In the end, the resulted pyramidal features are then fed into each of the heads, which yield their dense predictions.

Table 3 reflects the number of parameters each model has, and the number of **FL**oating Point **OP**erator**s** (FLOPs) it uses. Its aim is to underline that the MT-Net, and *MBMT-Net* variants have roughly the same number of parameters, which makes them easier to be compared. We also point out that in our training schedule, the number of parameters the *MBMT-Net*-CAT/-ADD/-MUL has have dramatically decreased (compared to the number of parameters of MT-Net) due to the pre-trained encoders. See experimental results in Table 6, that we interpret in Section VI. Furthermore, the number of FLOPs greatly varies for each of the compared models. It can be clearly noticed that of all the multi-task architectures, our *MBMT-Net* variants have the lowest number of FLOPs, and less than one eighth of the number of FLOPs of state-of-the-art MTAN [10].

B. TRAINING STAGE

Considering the multiple models we are working with, we had to be thorough in our training methodology. Moreover, for our research to be replicable, the step-by-step guild towards the full training setup is formulated in the following paragraphs.

We aimed to exhaustively test architecture variants with different hyper-parameters. Since the complexity of testing architectures against each other may affect the validity of the research, we had to perform multiple rounds of training, with different seeds.

The training of the models has been performed on a single Nvidia GPU. Due to hardware issues we faced, we had to adapt the EfficientPS implementation, so that the InPlace-ABN layers had to be converted to regular Batch-Norm layers [32], since the GPUs we trained the models on did not support the mapillary implementation [33]. Consequently, this increased the memory consumption, and decreased the efficiency by a bit.

As we have previously mentioned in Section IV, we have followed a cyclic process of improving our architecture. For that to be possible, we had to clearly determine the steps to follow in the process.

TABLE 3. Analysis of the number of parameters for the architectures involved in the experiments.

		Architecture						
		EfficientPS-SS	EfficientPS-DE	EfficientPS-SNP	MT-Net	*MBMT-Net*-CAT	*MBMT-Net*-ADD *MBMT-Net*-MUL	MTAN
Trainable	**#Parameters**	10M	11.2M	11.2M	32.6M	**8.7M**	**7.1M**	44.2M
	#FLOPs	3.251G	11.355G	11.36G	25.144G	**24.677G**	**24.213G**	0.218T
Total	**#Parameters**	10M	11.2M	11.2M	32.6M	**32M**	**30.4M**	44.2M
	#FLOPs	3.251G	11.355G	11.36G	25.144G	**24.677G**	**24.213G**	0.218T

Therefore, we decided to train ST-Nets on the three tasks of choice, then the MT-Net, and eventually our *MBMT-Net*. We must mention that we only present the remarkable, and interesting results in this paper, since many of the experiments have not yielded relevant results to us. However, among the performed experiments we count training all the architectures with different backbone layouts, ranging from EfficientNet-b0 to -b5, using a varying number of backbones in *MBMT-Net*, and freezing and unfreezing parts of the models.

The reason why these experiments did not make it in the paper is that the models could not have been well compared, because of either the number of parameter varying too much, or because the results met our expectations, and did not provide further insight into the architectures development.

That being said, we decided to fully train (i.e. no frozen layers) the ST-Nets. Then, we fully trained the MT-Net on all the tasks simultaneously. Both the pre-trained, and scratch backbone variants proved to yield the same performances. Eventually, when it came to *MBMT-Net*, we implemented our proposed training schedule, namely that we used the pre-trained backbones from each of the ST-Nets that we have frozen, and only trained the decoder part of the network till the loss flattened. Not only that this schedule dramatically reduced the GPU memory consumption, but it also increased the performance by a bit, compared to the fully trained *MBMT-Net*.

C. TESTING

After considering the training methodology, we are going to specify how we are going to perform the testing stage. Firstly, the pre-processed NYUv2 data set is split into 795 training and 654 testing samples, which gives us approximately a 55%-45% training-testing split. The same ratio, and testing methodology is employed in paper [10]. Similar to the results presented by Liu et al. [10], our models have not been trained on augmented data.

1) PERFORMANCE EVALUATION METRICS

The evaluation measures employed for assessing the performance of our *MBMT-Net* model on a training data set are the ones also considered by Liu et. al [10]. Because we address the same problem as they do, we are going to evaluate the models, by employing the same testing methodology that is used in [10], so our work is comparable to theirs. For easier comprehension of the tables, we use acronyms for the metrics that we will put forward.

Before introducing all the metrics, we must mention what each notation represents. On a certain testing data set, the confusion matrices for *nb_class* SS object classes are computed, where *nb_class* is the number of distinct semantic classes the NYUv2 data set has. As in any binary classification task, the confusion matrices consist of four values (*TP*, *FP*, *TN*, *FN*), where: *TP* represents the number of true positives, *TN* is the true negatives number, *FP* counts the false positives, and *FN* records the false negatives. These numbers are obtained from Furthermore, n is the total number of pixels that have a valid value in the ground truth mask, and all $pred_i$ (predicted pixels) and gt_i (ground truth pixels) are considered to be valid – where i is a pixel's index, considering the flattened image representation.

For each of the considered tasks (SS, DE, SNP) we further describe the performance metrics employed in the testing stage.

a: SEMANTIC SEGMENTATION

The metrics used for performance evaluation in SS tasks are:

- **Mean intersection over union** (*mIoU*)
  For each class $c \in nb\_class$, the *Intersection over Union* measure IoU_c is computed as shown in Formula (1).

$$IoU_c = \frac{TP}{TP + FP + FN} \cdot 100 \tag{1}$$

  Then, the *mIoU* measure is being obtained by taking the mean of the IoU_c values (Formula (2)). Thus, the values of *mIoU* range from 0 to 100, higher meaning better.

$$mIoU = \frac{\sum_{c \in nb\_class} IoU_c}{|nb\_class|} \tag{2}$$

- **Pixel accuracy** (*PAcc*)
  The *PAcc* uses the accuracy formula obtained from the corresponding confusion matrix. The metric for a single class is computed according to Formula (3).

$$PAcc_c = \frac{TP + TN}{TP + FP + TN + FN} \cdot 100 \tag{3}$$

  PAcc is the mean among all the considered classes. Consequently, it ranges from 0 to 100, and the higher it is, the better the model.

b: DEPTH ESTIMATION

The metrics used for performance evaluation in DE tasks are:

- **Absolute error** (*AbsErr*)
  The *Abs* represents the average of the absolute value of the differences between any two pixels on the same position of the predicted and ground truth dense masks, as given in Formula (4). Thus, the values of *AbsErr* range from 0 to 1, where 0 means the match is perfect between the predictions and the ground truth data.

$$AbsErr = \frac{1}{n} \cdot \sum_{i=1}^{n} |pred_i - gt_i| \tag{4}$$

- **Relative error** (*RelErr*)
  Similarly to *AbsErr*, *RelErr* is obtained by further dividing each of the sum terms of *AbsErr* by the ground truth value, as given in Formula (5). *RelErr* also ranges from 0 to 1, where lower is better.

$$RelErr = \frac{1}{n} \cdot \sum_{i=1}^{n} \frac{|pred_i - gt_i|}{gt_i} \tag{5}$$

c: SURFACE NORMAL PREDICTION

All the measurements used for performance evaluation of the SNP task are computed from the base error, which is defined in Formula (6). First, the element-wise multiplication between the *prediction* and *ground truth* matrices is computed (the content of the innermost parenthesis). We are considering all the RGB pixels in one image that have a valid equivalent in the ground truth mask. Then, these values are clamped to the [-1,1] interval, so that the *arc cosine* function is well defined on the interval. Then, the result of the arc cosine is converted into *degrees*. Thus, assuming that i is the index of a pixel (that has a valid value in the ground truth mask) in the flattened image representation, we are computing the error Err_i as shown Formula (6).

$$Err_i = degrees\left(\arccos\left(clamp_{[-1,1]}\left(pred_i \cdot gt_i\right)\right)\right) \tag{6}$$

The following metrics are employed for performance evaluation in SNP tasks:

- **Mean angle distance** (*Mean*)
  The value of this metric is computed as shown in Formula (7). Because of the conversion to degrees, and the fact that on a plane we can see only half of the possible angle values, *Mean* ranges from 0 to 180, lower values suggesting a better performance.

$$Mean = \frac{1}{n} \cdot \sum_{i=1}^{n} Err_i \tag{7}$$

- **Median angle distance** (*Med*)
  Med is computed according to Formula (8)

$$Med = \begin{cases} sErr_{(n+1)/2} & n \bmod 2 = 1; \\ \dfrac{sErr_{n/2} + sErr_{n/2+1}}{2} & \text{otherwise} \end{cases} \tag{8}$$

  where $sErr_n$ is the n-th element of the increasingly sorted, flattened matrix. The value range of *Med* is 0 to 180, where lower means better performance.

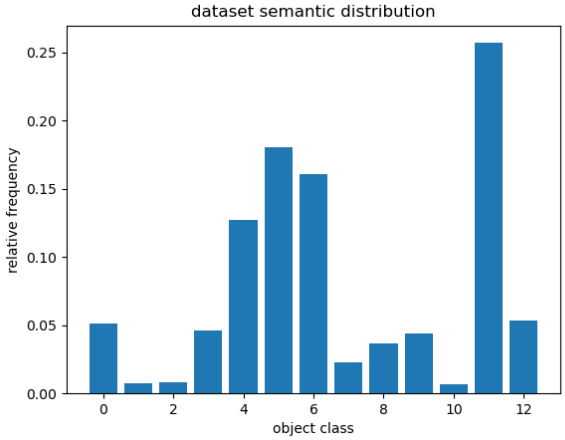

FIGURE 4. NYUv2 semantic label distribution.

- **Mean number of angles** having an error less than α (MNA_α)
  Considering that α is an error threshold measured in degrees, the mean number of angles having an error less than α is computed as in Formula (9).

$$MNA_\alpha = \frac{1}{n} \cdot \sum_{i=1}^{n} count(Err_i, \alpha) \cdot 100 \tag{9}$$

  The values of *MNA* range from 0 to 100, since it represents the proportion of predicted angles having a lower error than α. Therefore, higher is better. Additionally, the values for the *count* function are computed as shown in Formula (10).

$$count(x, \alpha) = \begin{cases} 1 & x < \alpha; \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

V. DATA SET AND EXPERIMENTAL SETUP

The data set and the experimental setup employed for the experimental evaluation of the *MBMT-Net* model are further presented.

A. DATA SET

Our data set of choice is NYUv2, provided by [10]. It consists of 1449 288 × 384 raw RGB images of indoors scenes. We use them as input for the models, and their corresponding segmentation, depth, and surface normal masks as ground truth data. Due to its moderate size, we were able to exhaustively perform experiments with various model configurations to support our proof of concept, and decide on what substantially improved the performance of the three tasks (SS, DE, SNP).

Figures 4 and 5 depict the distributions of SS and DE labels, respectively, in the whole NYUv2 data set provided by [10]. The test and train distributions slightly vary, but they closely follow the same patterns. We have not considered invalid pixels in the data set, because it would have skewed the results.

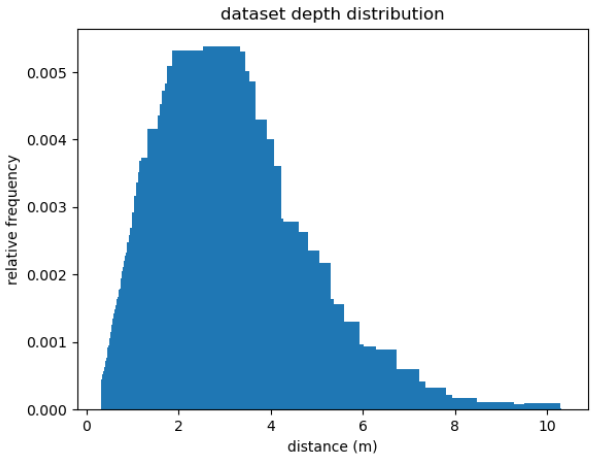

FIGURE 5. NYUv2 depth values distribution.

Figure 4 presents us the semantic classes distribution, namely, from 0 to 12, *bed*, *books*, *ceiling*, *chair*, *floor*, *furniture*, *objects*, *pict./deco*, *sofa*, *table*, *TV*, *wall*, and *window*. As expected, walls, the floor, furniture and different sorts of objects make up the vast majority of semantically labeled items. Considering that the less structurally complex, and better described objects, such as beds, sofas, chairs, windows or TVs are not that often seen in the NYUv2 data set scenes, we may think that the class imbalance, especially for the class *objects*, may negatively influence the overall semantic segmentation results. This would happen because there are too many types of household items labelled as *objects*, while they have no relevant similarities between them, therefore making it difficult for the ML models to learn. Nonetheless, if different architectures could reliably predict items from the less frequent group, it would certainly be a remarkable, and robust one, with great potential for solving difficult class imbalance problems.

As we can notice from Figure 5, most of the indoors scenes have depth values in the 1 to 5 meters range. However, rarely, there are isolated instances larger than 6 meters, which could be explained by the existence of large rooms, or hallways.

B. EXPERIMENTAL SETUP

The experiments have been performed on a single Nvidia GPU. We trained the model until the loss flattened out. Usually, this took less than the pre-set number of epochs but we used 700 for consistency.

As previously shown in Section IV, a grid search was performed for hyper-parameters optimisation. The hyper-parameters considered in the grid search are presented in Table 4, with their final values we have used in our experiments.

The ST networks train both their encoder and decoder. Afterwards, we use transfer learning for *MBMT-Net*, so that it benefits from the pre-trained EfficientNet-b2 backbones. The reported results refer to the *MBMT-Net* having frozen its encoders during training. During experiments, we have tried both pre-trained and cold start trainings and the results did not improve having the encoders also involved in the process.

TABLE 4. Optimal training parameters resulted from the grid search.

Training Configuration	ST-Net	MT-Net	*MBMT-Net*
number of epochs	700	700	700
batch size	8	8	8
learning rate	1e-5	1e-5	1e-5
DWA temperature	-	2.0	2.0
number of backbones	1	3	3

For the MT-Nets, we implemented the DWA procedure presented in [10] and we set the *temperature* parameter to 2.0. The temperature parameter controls the softness of task weighting, meaning that a higher temperature would assign equal importance to all the performed tasks. Liu et al. [10] empirically found that 2.0 is the optimum value across all tasks. The rest of the training procedure is identical to the one used in [10]. The results are compared to MTAN DWA trained on the three tasks as well.

In what concerns the losses, SS is learned using Depth-Wise Cross Entropy, DE is optimised using L1 Norm and for SNP the Dot Product between the predictions and ground truth labels is considered. Furthermore, all the losses consider *binary pixel validity masks* for evaluating the differences only of the pixels that have a valid corresponding ground truth label.

VI. RESULTS AND DISCUSSION

In this section we are going to highlight the main findings of our experiments carried out with the goal of evaluating the performane of the *MBMT-Net* model introduced in Section IV. First, an ablation study is conducted in Section VI-A in order to determine the best performing *MBMT-Net* architecture. Then, Section VI-B presents the results of our final *MBMT-Net* architecture compared to the state-of-the-art model for multi-task learning [10]. An interpretation of our results and comparison to related work are described in Section VI-C. Extensive implementation details may be seen in the code available on the *MBMT-Net* GitHub repository [34].

A. ABLATION STUDY

This section presents our line of reasoning that led to our final *MBMT-Net* architecture.

While our approach may not be scalable enough to allow many more backbones, it is the best we have experimented with. This is due to the fact that addition (ADD) and multiplication (MUL) feature map fusion techniques extend their domain of values by severalfold in relation to the number of backbones. Although concatenation (CAT) results in more FLOPs being used in the two-way FPN, the output of the *MBMT-Net* backbone has the same shape as the one of the MT-Net. As a result, concatenation scalability does not affect the parsing heads' runtime performance. Even though the number of parameteres and FLOPs for ADD/MUL is lower

TABLE 5. Ablation study of several *MBMT-Net* variants.

MT Variants Default: -CAT	SS Loss	mIoU	Pacc	DE Loss (AbsErr)	RelErr	SNP Loss	Angle Distance		MNA_α		
							Mean	Med	$\alpha = 11.25$	$\alpha = 22.5$	$\alpha = 30$
MB2-xDE	2.1844	14.86	50.60	0.6478	0.2822	**0.2150**	32.4459	27.8443	19.40	41.04	53.41
MB2-xSNP	2.1985	14.19	49.22	0.6700	0.2734	0.2329	34.3880	30.4422	16.30	36.92	49.32
MB2-xSS	2.2052	12.22	48.29	0.6667	0.2905	0.2169	32.7242	28.2637	18.58	40.25	52.81
MB3 (cold)	2.2314	10.95	45.64	0.7702	0.3165	0.2518	36.7015	34.1046	12.19	31.15	43.60
MB3-ADD	2.2142	12.15	47.44	0.6581	0.2711	0.2283	33.7848	29.4739	17.54	38.53	50.88
MB3-MUL	2.2380	11.35	45.02	0.7039	0.2996	0.2489	36.0989	32.7465	13.76	33.28	45.69
MB3 (pre)	**2.1605**	**24.84**	**52.76**	**0.6076**	**0.2569**	0.2174	**32.0937**	**26.3163**	**21.69**	**43.66**	**55.62**

than for CAT (Table 3), the range of values greatly increases. The resulting feature maps' values are going to dramatically increase, up to the point of overflowing the data type representations, therefore resulting in severe performance issues. This underlines the fact that ADD and MUL approaches are not scalable with the number of backbones, and with the encoders' complexity.

Table 5 aims to underline our contribution regarding the structural elements of the final *MBMT-Net* architecture. We evaluate the models when we eliminate backbones, the pre-training step, both, or change backbone output fusion strategy. The best performance is highlighted in green, and the second best one is colored in yellow.

Firstly, we note that the first column is colored with blue or red, meaning the training has been done from a cold start or pre-trained, respectively. Secondly, not to overcomplicate the notations, the default feature maps merging operation is concatenation (CAT) unless specified otherwise. Besides the color hints, we differentiate the MB3 variant training schedule by specifying whether it was trained from a cold start (cold), or it was pre-trained (pre).

The last three rows of Table 5 aim to help us select the most suitable feature merging operation. As depicted, the *MBMT-Net* variants employing addition or multiplication as backbone output fusion severely underperform. The results confirm our intuition that algebraically merging the feature maps is an ill-posed problem, especially when varying the number of backbones. Consequently, the rest of the experiments are performed using channel-wise concatenation (CAT).

Subsequently, the top first three rows present the results of the experiments performed using *MBMT-Net* without one of the three pre-trained backbones. It is notable that even though the backbones do not take part in the whole learning process of the model, the missing features are essential. For training the decoder part of the network, it appears that the decreased number of parameters dramatically affects the overall performance. Additionally, by covering all scenarios for the two backbones *MBMT-Net*, we can clearly agree that the number of pre-trained backbones is suitable for the chosen tasks, and it effectively helps the model learn useful features.

Then, in order to prove the importance of the training schedule (pre-training the backbones especially), the rows nominating MB3 colored in blue and red compare training schedules impact on the performance of the final *MBMT-Net*-CAT architecture (the architecture depicted in the last row from Table 5). Hence, we notice that the *MBMT-Net* trained from a cold start considerably underperforms. That may happen because of the architecture's inability to propagate much larger or smaller gradients into each of the backbones. This results in the encoders learning about the same feature maps, which defeats the purpose of the multi-backbone model. Nonetheless, if we were to train the decoder part of the network according to our proposed schedule (encoder pre-training, freezing, decoder training), the results are a lot better.

B. RESULTS

As a result of the previously conducted ablation study, the final *MBMT-Net* architecture variant which performs the best is the one that employs the CAT feature maps merging operation.

Table 6 summarizes the results for ST-Nets, MT-Net, *MBMT-Net* variants, and the SOTA MTAN model [10]. While all the ST-Nets are in the same EfficientPS column, their architecture and number of parameters differ as explained in Section IV-A and previously mentioned in Table 3. There are missing values for the MTAN tasks loss, as they have not been reported by Liu et al. [10].

One of the first findings is that the MT-Net is outperformed by the ST-Nets in SS and SNP regarding not only the losses, but also the rest of the metrics. This may be explained by the appearance of oversaturation of the single backbone, which cannot generalize well for all the tasks at once. However insignificant the differences may seem, it is to be noted that the ST-Nets employ the EfficientNet-b2 compared to the MT-Net which uses the EfficientNet-b5. The difference of encoder parameters is almost 17.4M more in the case of MT-Net. Needless to say, all the trainings have been done until the loss flattened and multiple checkpoints have been evaluated, yielding similar results, with little variance.

Considering the aforementioned arguments, our decisions in developing this model building strategy can be regarded as favourable for achieving robust and effective networks.

Therefore, our *MBMT-Net* approach proves superior to the traditional single-backbone MT-Net, completely outperforming it and every ST-Net. Thanks to the multiple pre-trained backbones that provide the network with more diverse features, we may conclude that the feature extraction means become more robust. Consequently, the concatenated

TABLE 6. Results of the architectures described in Section IV compared to state-of-the-art MTAN model [10] on the three tasks of choice.

Task	Measure	Architecture			
		EfficientPS	MT-Net	*MBMT-Net*-CAT	MTAN (SOTA)
Semantic Segmentation	Loss	2.1877	2.1990	**2.1605**	-
	mIoU	21.23	20.87	**24.84**	17.15
	PAcc	50.06	48.88	52.76	**54.97**
Depth Estimation	Loss	0.6861	0.6185	**0.6076**	-
	AbsErr	0.6861	0.6185	0.6076	**0.5906**
	RelErr	0.2913	0.2646	**0.2569**	**0.2569**
Surface Normal Prediction	Loss	0.2249	0.2358	**0.2174**	-
	Mean	32.8582	33.9981	32.0937	**31.60**
	Med	27.2290	28.7907	26.3163	**25.46**
	$MNA_{11.25}$	20.49	18.81	21.69	**22.48**
	$MNA_{22.5}$	42.22	39.75	43.66	**44.86**
	MNA_{30}	54.22	51.88	55.62	**57.24**

condensed representation is infused with pattern detection capabilities of the three tasks, which complements the insufficiency of the others to correctly make predictions. Furthermore, our implementation uses less parameters than MT-Net, which implies there is no additional complexity to support the improvement in this regard.

By designing such MB-MT architecture, we manage to compete with MTAN [10] and achieve results close to the best for SNP, but still lacking at most 2 percentages in all the measurements. Nevertheless, the DE performance reaches equal values for *RelErr* and a little worse for *AbsErr*. Finally, we are able to surpass the state-of-the-art MTAN *mIoU* value by a considerably wide margin of 7.69%. This may be explained by the sheer processing potential of the chosen EfficientPS architecture, as we can see any of our architecture variants have better SS *mIoU*-s. Additionally, by infusing our concatenated feature maps with the complementary pattern recognition means, we successfully increase the baseline performance of the ST-Net by 3.61% and by 3.97% when compared to the MT-Net. Moreover, we emphasize on the importance of employing multiple separated backbones. The specifically trained encoders can substantially increase the performance and overcome the *oversaturation* problem. Considering that, our architecture design strategy shall act as a robustness booster, as the diverse features may act as a regulariser in the multi-task context.

Overall, our model did not consistently outperform MTAN. However, considering the 12.16M parameters deficit, the *MBMT-Net* architecture proved to effectively benefit from our design choices. Nevertheless, further experiments using larger multi-backbones have to be performed to conclude the findings.

A more elaborated side-by-side comparison between MT-Net and *MBMT-Net* can be observed in Figure 6. There we have five sample images from NYUv2. For each image we have considered displaying the raw image (Raw), ground truth masks (GT), *MBMT-Net* prediction and MT-Net prediction. From top to bottom, the tasks are semantic segmentation (SS), surface normal prediction (SNP), and depth estimation (DE). If we take a closer look into the plots, we can easily notice some differences between the qualitative performance of the two models. First, considering semantic segmentation, we can easily see that our model, *MBMT-Net*, preserves the shape consistency more than MT-Net. While for MT-Net, all of the predicted SS masks contain irregular spots in the middle of the objects, *MBMT-Net* tends to better understand the scene composition, and aggregate parts of the objects together. Furthermore, this fact can be also noticed in the SNP predicted masks. Not only that the shape consistency is clearly visible, but *MBMT-Net* also offers more distinguishable depth of field, preserving the volume of objects, and more reliably reconstructing the surfaces. Another important aspect is that NYUv2 has a lot of unlabelled pixels, which challenges the learning capabilities of our models. However, from what we can see in the SNP plots, *MBMT-Net* can better generalise than MT-Net. To support this statement, we look at the structural integrity of objects. More precisely, the edges and corners of sharper objects are more prominent, and better represented. Thanks to the general knowledge of the environment provided by the pre-trained backbones, our *MBMT-Net* benefits from, it unarguably better distinguishes thinner objects, and it respects the space between objects, without blurring or filling it, as MT-Net does (e.g. table legs). Moreover, these characteristics offer *MBMT-Net* its robustness, and consistency across multiple tasks (i.e. it can be seen that all the structures from SS appear in SNP and DE, and vice-versa. However, MT-Net is inconsistent, as some features flicker across the performed tasks.

C. DISCUSSION

Yudong et al. [11] are also using multiple backbones in their proposed CBNet architecture, as in our proposal. However, our approach differs from the previous one [11], as we aim to allow any combination of processing units in the learning context, not just identical structures. Our additional goal was to preserve the number of multi-backbones parameters similar to the one of a traditionally structured MT model, without loosing performance.

Although a quantitative comparison cannot be conducted, as our approached tasks are different from the ones

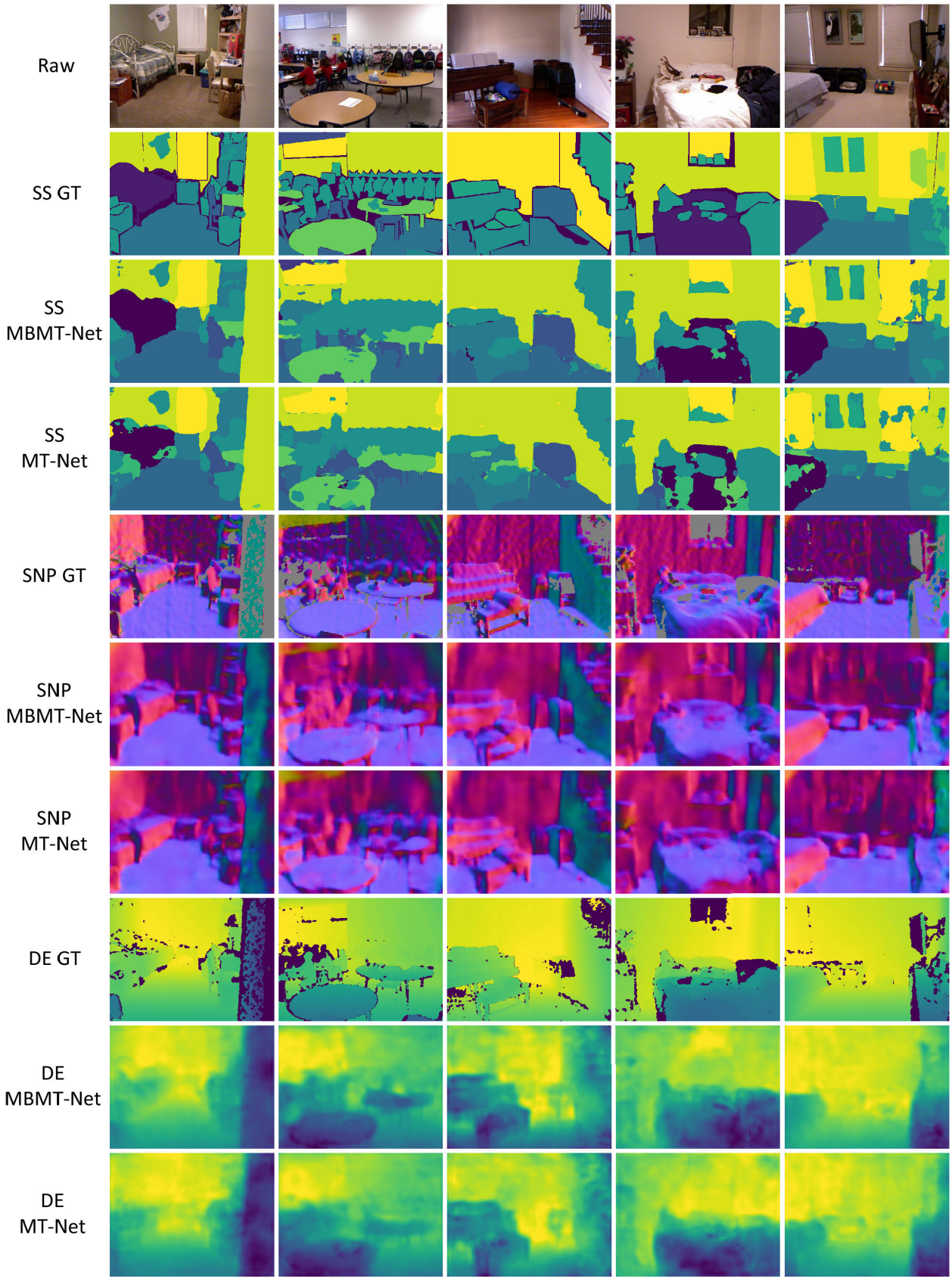

FIGURE 6. MT-Net vs *MBMT-Net* comparison on the three tasks of choice.

TABLE 7. Relative improvements of *MBMT-Net* with respect to MT-Net and two other state-of-the-art architectures (MTAN and EfficientPS), considering all the proposed metrics.

Improvement *MBMT-Net*	# Parameters		SS Loss	mIoU	PAcc	DE Loss	AbsErr	RelErr	SNP Loss	Angle Distance		MNA_α		
	Trainable	Total								Mean	Med	$\alpha = 11.25$	$\alpha = 22.5$	$\alpha = 30$
vs MTAN [10]	80.31%	27.6%	n/a	44.84%	-4.02%	n/a	-2,.8%	0.00%	n/a	-1.56%	-3.36%	-3.51%	-2.67%	-2.83%
vs EfficientPS [19]	n/a	n/a	1.24%	17.00%	5.39%	11.44%	11.44%	11.81%	3.33%	2.33%	3.35%	5.86%	3.41%	2.58%
vs MT-Net	73.31%	1.84%	1.75%	19.02%	7.94%	1.76%	1.76%	2.91%	7.80%	5.60%	8.59%	15.31%	9.84%	7.21%

CBNet [5], [11] tackled, the contributions regarding the backbones' structure can be qualitatively compared. Instead of adopting CBNet communication scheme that resembles a multilayer perceptron, we have no communication between the backbones till the very end. Thus, we manage to keep the number of parameters and FLOPs relatively low compared to their work. Furthermore, our approach is better scalable with the depth of the backbones, while the CBNet scheme increases exponentially with the number of layers, and with the number of used backbones. Additionally, while their scheme is restrictive, and the backbones must have the shape, our structure allows for any combination of architectures. This makes us believe that our work is more suitable for deeper models, especially in multi-task learning contexts.

While MTAN [10] is the state-of-the-art for multi-task learning, EfficientPS represents the state-of-the-art for panoptic segmentation. The architecture itself makes use of one backbone, and two parsing heads that are used to predict a single, complex task. If we were to consider the de-merged output of its two heads, we can also look upon the architecture from a multitask learning perspective. That being said, EfficientPS can be regarded as the newer SOTA in multitask learning.

On top of the EfficientPS architecture, we have build our *MBMT-Net* specifically for performing the same tasks as MTAN. Moreover, our contribution consists of analysing and combining the two approaches, so that the DWA ensures better task convergence, the stronger Two-Way FPN extracts richer features, the two-step training schedule specializes our own multi-backbone architecture, and the concatenation feature maps merging operation yields the best results, while also being the most scalable approach.

We can say that according to our results, having multiple feature maps of the same input proved to be a reliable and robust source of information for the multi-task network. Since every pre-trained backbone outputs a different condensed representation for each task, by channel-wise concatenating them it is emphasized that the features are now more consistent. This results in robust and reliable sources, therefore significantly increasing the performance when jointly used in multi-task models for solving dense prediction tasks.

Table 7 captures the *MBMT-Net* improvements relative to the other state-of-the-art architectures. Let us consider that for the performance metric P we denote by P_{our} the value of P provided by our *MBMT-Net* model and by P_{other} the value of P provided by another architecture (MTAN) [10], EfficientPS, [19], MT-Net). The improvement *impr* (expressed as a percentage) achieved by our *MBMT-Net* model in terms of the performance measure P is computed as in Formula (11):

$$impr = \begin{cases} 100 \cdot \dfrac{P_{other} - P_{our}}{P_{other}} & \text{if } P \text{ has to be minimised;} \\ 100 \cdot \dfrac{P_{our} - P_{other}}{P_{other}} & \text{otherwise} \end{cases} \tag{11}$$

On the one hand, it is clearly visible that the traditional MT-Net is inefficient in infusing the shared backbone with relevant feature extraction means. Taking that into account, with a 1.84% improvement in parameters, *MBMT-Net* scores higher than MT-Net, especially regarding *mIoU*, where the improvement is of 19.02%. On the other hand, MTAN still represents a tough competitor to our model, regarding overall performance. However, by looking at the total parameter count improvement of 26.32%, we can confidently say that *MBMT-Net* has the capacity to compete with computationally expensive architectures even when having notably less parameters.

Moreover, our approach may prove useful especially in DAI contexts, since each backbone and each head may be instantiated on a separated GPU. That would facilitate the processing of more information at a time, without additional latency. As each backbone uses less parameters now, the overall multi-backbone processing time would be less than the one of a single and larger backbone in a DAI context. Additionally, each head could perform the processing in a non-sequential schedule, only the scatter and gather operations of sharing the feature maps between the GPUs representing the bottleneck in regards of hardware-specific bandwidth.

For verifying if the improvement achieved by our *MBMT-Net* model is statistically significant, a one tailed paired Wilcoxon signed-rank test [35], [36] was applied. The sample of performance metrics values obtained by the *MBMT-Net* model has been tested against the sample of values obtained by the other architectures (MTAN, EfficientPS, MT-Net). A p-value less than 0.01 was obtained, showing that the improvement achieved by *MBMT-Net* is statistically significant, at a significance level of $alpha = 0.01$.

VII. CONCLUSION AND FUTURE WORK

To summarise, we have introduced *MBMT-Net*, an alternative to the traditional hard parameter sharing multi-task model architecture. *MBMT-Net* consisted of parallel pre-trained backbones whose outputs are concatenated and offered to the MT heads. By doing so, the decoders benefit from richer and more diverse features with decreased network parameters when compared to traditional MT architectures. Our strategy is architecture independent, and it can be applied to different types of backbones and parsing heads, which greatly extends

the domain of configurable features, finally enhancing existing Single- and Multi-Task model building strategies and outperforming them when using the Multi-Backbone design.

First of all, we have shown how pre-training less complex backbones and putting them together offers better performance than using a single larger backbone. Second of all, we mentioned that our training schedule offers the possibility to train the models on lower-end GPUs, since freezing the backbones dramatically decreases the memory consumption, without trading off performance. Eventually, as an answer to **RQ1**, it turned out that *MBMT-Net* architectural style increases the performance of each evaluated task when compared to both the ST and single backbone MT networks, without supplementary parameter complexity. Furthermore, the triple backbones pre-training proved to be greatly beneficial both quantitatively and qualitatively, as Figure 6 shown us. Moreover, **RQ2** is being answered by having achieved scores comparable to previous models' results. Additionally, we have set a new state-of-the-art performance in multi-task learning, as *MBMT-Net* outperforms previously best models, while having a significant parameter deficit.

The last research question, namely **RQ3** is affirmatively answered by having performed the Wilcoxon signed-rank test, which confirms our improvement compared to the other models.

Regarding future improvements, we consider that the *MBMT-Net* model introduced in this paper represents a proof of concept. Its aim is to emphasize the multi-backbone importance, and motivate further research in this field of multi-task learning. This would eventually lead our real-time computer vision systems to extend their processing capacity to another level, allowing for powerful fine-grained feature extractors that may grant vehicle autonomy.

Besides that, in a DAI context, our model building technique would be of great interest for reducing latency while processing more data, which could be seen as a good starting point for real-time performance systems development.

ACKNOWLEDGMENT

The work of George Ciubotariu was supported by Babeș-Bolyai Univeristy through the Special Scholarship for Scientific Activity (2021-2022). The authors would like to thank the editor and the anonymous reviewers for their useful suggestions and comments that helped to improve the paper and the presentation.

REFERENCES

[1] A. Osipov, V. Shumaev, A. Ekielski, T. Gataullin, S. Suvorov, S. Mishurov, and S. Gataullin, "Identification and classification of mechanical damage during continuous harvesting of root crops using computer vision methods," *IEEE Access*, vol. 10, pp. 28885–28894, 2022.

[2] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, and A. Smola, "ResNeSt: Split-attention networks," 2020, *arXiv:2004.08955*.

[3] M. Yin, Z. Yao, Y. Cao, X. Li, Z. Zhang, S. Lin, and H. Hu, "Disentangled non-local neural networks," in *Proc. Eur. Conf. Comput. Vis.* (Lecture Notes in Computer Science), vol. 12360, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds. Glasgow, U.K.: Springer, 2020, pp. 191–207, doi: 10.1007/978-3-030-58555-6_12.

[4] J. Botsch, H. Jain, and O. Hellwich, "IMD-Net: A deep learning-based icosahedral mesh denoising network," *IEEE Access*, vol. 10, pp. 38635–38649, 2022.

[5] T. Liang, X. Chu, Y. Liu, Y. Wang, Z. Tang, W. Chu, J. Chen, and H. Ling, "CBNet: A composite backbone network architecture for object detection," 2021, *arXiv:2107.00420*.

[6] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6230–6239, doi: 10.1109/CVPR.2017.660.

[7] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, 2019, pp. 558–567. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/He_Bag_of_Tricks_for_Image_Classification_with_Convolutional_Neural_Networks_CVPR_2019_paper.html

[8] M. Crawshaw, "Multi-task learning with deep neural networks: A survey," 2020, *arXiv:2009.09796*.

[9] P. K. N. Silberman, D. Hoiem, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis.* (Lecture Notes in Computer Science), vol. 7576, A. W. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Florence, Italy: Springer, 2012, pp. 746–760, doi: 10.1007/978-3-642-33715-4_54.

[10] S. Liu, E. Johns, and A. J. Davison, "End-to-end multi-task learning with attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, 2019, pp. 1871–1880. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Liu_End-To-End_Multi-Task_Learning_With_Attention_CVPR_2019_paper.html

[11] Y. Liu, Y. Wang, S. Wang, T. Liang, Q. Zhao, Z. Tang, and H. Ling, "CBNet: A novel composite backbone network architecture for object detection," in *Proc. 34th AAAI Conf. Artif. Intell., 32nd Innov. Appl. Artif. Intell. Conf. (IAAI), 10th AAAI Symp. Educ. Adv. Artif. Intell. (EAAI)*, New York, NY, USA, 2020, pp. 11653–11660. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/6834

[12] U. Sehar and M. L. Naseem, "How deep learning is empowering semantic segmentation," *Multimedia Tools Appl.*, vol. 81, no. 21, pp. 30519–30544, Apr. 2022, doi: 10.1007/s11042-022-12821-3.

[13] S. Hao, Y. Zhou, and Y. Guo, "A brief survey on semantic segmentation with deep learning," *Neurocomputing*, vol. 406, pp. 302–321, Sep. 2020, doi: 10.1016/j.neucom.2019.11.118.

[14] F. Khan, M. A. Farooq, W. Shariff, S. Basak, and P. Corcoran, "Towards monocular neural facial depth estimation: Past, present, and future," *IEEE Access*, vol. 10, pp. 29589–29611, 2022.

[15] G. Ciubotariu, V. Tomescu, and G. Czibula, "Enhancing the performance of image classification through features automatically learned from depth-maps," in *Proc. Int. Conf. Comput. Vis. Syst.* (Lecture Notes in Computer Science), vol. 12899, M. Vincze, T. Patten, H. I. Christensen, L. Nalpantidis, and M. Liu, Eds. Berlin, Germany: Springer, 2021, pp. 68–81, doi: 10.1007/978-3-030-87156-7_6.

[16] Y. Ming, X. Meng, C. Fan, and H. Yu, "Deep learning for monocular depth estimation: A review," *Neurocomputing*, vol. 438, pp. 14–33, May 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220320014

[17] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 2650–2658, doi: 10.1109/ICCV.2015.304.

[18] X. Wang, D. F. Fouhey, and A. Gupta, "Designing deep networks for surface normal estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 539–547, doi: 10.1109/CVPR.2015.7298652.

[19] R. Mohan and A. Valada, "EfficientPS: Efficient panoptic segmentation," *Int. J. Comput. Vis.*, vol. 129, pp. 1551–1579, Feb. 2021, doi: 10.1007/s11263-021-01445-z.

[20] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

[21] L. Chen, M. D. Collins, Y. Zhu, G. Papandreou, B. Zoph, F. Schroff, H. Adam, and J. Shlens, "Searching for efficient multi-scale architectures for dense image prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Montreal, QC, Canada, 2018, pp. 8713–8724. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/c90070e1f03e982448983975a0f52d57-Abstract.html

[22] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," 2018, *arXiv:1808.03314*.
[23] S. M. Bafti, S. Chatzidimitriadis, and K. Sirlantzis, "Cross-domain multitask model for head detection and facial attribute estimation," *IEEE Access*, vol. 10, pp. 54703–54712, 2022.
[24] P. Vafaeikia, K. Namdar, and F. Khalvati, "A brief review of deep multitask learning and auxiliary task learning," 2020, *arXiv:2007.01126*.
[25] A. Amyar, R. Modzelewski, H. Li, and S. Ruan, "Multi-task deep learning based CT imaging analysis for COVID-19 pneumonia: Classification and segmentation," *Comput. Biol. Med.*, vol. 126, Nov. 2020, Art. no. 104037. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482520303681
[26] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017, *arXiv:1706.05098*.
[27] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Proc. Adv. Neural Inf. Process. Syst.*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Montreal, QC, Canada, 2015, pp. 2962–2970. [Online]. Available: https://proceedings.neurips.cc/paper/2015/hash/11d0e6287202fced83f79975ec59a3a6-Abstract.html
[28] M. Feurer, K. Eggensperger, S. Falkner, M. Lindauer, and F. Hutter, "Auto-Sklearn 2.0: Hands-free AutoML via meta-learning," 2020, *arXiv:2007.04074*.
[29] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and A. Desmaison, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alche-Buc, E. B. Fox, and R. Garnett, Eds. Vancouver, BC, Canada, 2019, pp. 8024–8035. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html
[30] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Mach. Learn. Res.*, vol. 97, K. Chaudhuri and R. Salakhutdinov, Eds. Long Beach, CA, USA, 2019, pp. 6105–6114. [Online]. Available: http://proceedings.mlr.press/v97/tan19a.html
[31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.
[32] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, vol. 37, F. R. Bach and D. M. Blei, Eds. Lille, France, 2015, pp. 448–456. [Online]. Available: http://proceedings.mlr.press/v37/ioffe15.html
[33] S. R. Bulo, L. Porzi, and P. Kontschieder, "In-place activated BatchNorm for memory-optimized training of DNNs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 5639–5647. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Bulo_In-Place_Activated_BatchNorm_CVPR_2018_paper.html
[34] *MBMT-Net Github Repository*. Accessed: Nov. 15. [Online]. Available: https://github.com/george200150/MBMT-Net/
[35] S. Siegel and N. Castellan, *Nonparametric Statistics for the Behavioral Sciences*, 2nd ed. New York, NY, USA: McGraw-Hill, 1988.
[36] *Social Science Statistics*. Accessed: Jun. 15. [Online]. Available: http://www.socscistatistics.com/tests/

GEORGE CIUBOTARIU is currently pursuing the M.Sc. degree in applied computational intelligence specialization from the Faculty of Mathematics and Computer Science, Babeş-Bolyai University, Cluj-Napoca, Romania. He is also a Bosch Computer Vision Engineer working with the Engineering Center Cluj. His research interests include computer vision and machine learning.

GABRIELA CZIBULA works as a Professor at the Computer Science Department, Faculty of Mathematics and Computer Science, Babeş-Bolyai University, Cluj-Napoca, Romania. She has published more than 200 papers in prestigious journals and conferences proceedings. Her research interests include machine learning, distributed artificial intelligence and multiagent systems, and bionformatics.