Neural Architecture Search in the Context of Deep Multi-Task Learning

Guilherme Gadelha[1][a], Herman Gomes[1][b] and Leonardo Batista[2][c]
[1]*Federal University of Campina Grande, Brazil*
[2]*Federal University of Paraiba, Brazil*

Keywords: Multi-Task Learning, Neural Architectural Search, Reinforcement Learning, Deep Learning.

Abstract: Multi-Task Learning (MTL) is a neural network design paradigm that aims to improve generalization while simultaneously solving multiple tasks. It has obtained success in many application areas such as Natural Language Processing and Computer Vision. In an MTL neural network, there are shared task branches and task-specific branches. However, automatically deciding on the best locations and sizes of those branches as a result of the domain tasks remains an open question. With the aim of shedding light to the above question, we designed a sequence of experiments involving single-task networks, multi-task networks, and networks created with a neural architecture search (NAS) strategy. In addition, we proposed a competitive neural network architecture for a challenging use case: the ICAO photograph conformance checking for issuing of passports. We obtained the best results using a handcrafted MTL network, whose effectiveness is close to state-of-the-art methods. Furthermore, our experiments and analysis pave the way to develop a technique to automatically create branches and group similar tasks into an MTL network.

1 INTRODUCTION

Single-Task Learning (STL) is a traditional learning paradigm in which a neural network is defined and meticulously tuned for solving a specific task (Caruana, 1997). However, manually finding and tuning a neural network is time-consuming, and early research indicated that performance might increase if some tasks were solved together (Caruana, 1997), giving rise to the so-called Multi-Task Learning (MTL) paradigm. MTL networks present a lower memory footprint and higher inference speeds, when compared to equivalent solutions based on STL (Vandenhende et al., 2021). Recent reviews (Ruder, 2017; Zhang and Yang, 2021; Vandenhende et al., 2021) have identified different strategies for designing these networks in a partial or fully automatic way.

MTL is a general learning paradigm that aims to improve the generalization performance of *related tasks* compared to the results achieved in isolation (STL) (Caruana, 1997). In MTL, the tasks are learned in parallel and share common representations, and may operate as regularizers for one another (Vandenhende et al., 2021).

There is also an increasing interest on Neural Architecture Search (NAS) applied to MTL. NAS focuses on automatic techniques for neural architecture design, where layers or blocks are searched and arranged in different ways within the network architecture. Candidate architectures are tested against a validation dataset until convergence, when the best architecture is selected. There are some strategies to explore the architecture search space, such as Random Search, Bayesian Optimization, Evolutionary Methods, Reinforcement Learning (RL), and Gradient-based Methods (Elsken et al., 2019). Random Search is usually applied as a baseline to evaluate the proposed strategies, as we also do in this research. NAS methods based on RL (Zoph and Le, 2017; Pham et al., 2018) were responsible for the popularization of the field.

In this paper we evaluate handcrafted Single-Task Learning and Multi-Task Learning architectures, as well as architectures discovered through NAS for a challenging problem: the International Civil Aviation Organization (ICAO) photographic compliance checking. Each approach has been evaluated in terms of Equal Error Rate (EER). The objective of the experiments is to give insights for partial or fully automated design of network architectures.

The ICAO ISO/IEC 19795-4 specification (ICAO, 2015) defines the main components of biometric identification systems. In particular, for facial identifica-

[a] https://orcid.org/0000-0002-1426-2090
[b] https://orcid.org/0000-0002-0208-2041
[c] https://orcid.org/0000-0002-1069-3002

Gadelha, G., Gomes, H. and Batista, L.
Neural Architecture Search in the Context of Deep Multi-Task Learning.
DOI: 10.5220/0011696200003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 5: VISAPP*, pages 684-691
ISBN: 978-989-758-634-7; ISSN: 2184-4321

tion, the ISO/IEC 19794-5 (ISO, 2017) standard proposes best practices for face photographs, including 23 compliance requisites. Figure 1 contains some examples of facial images with issues (the three images on the left) and one example that complies (the last image on the right) with some of the ICAO requisites.

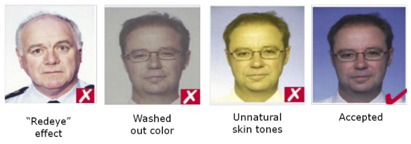

Figure 1: Examples of images not complaint and complaint with some ICAO requisites.

The main contributions of this work are (i) the proposition of a new NAS-MTL technique, (ii) a comparative study between handcrafted STL, handcrafted MTL, and NAS, and (iii) obtaining a competitive result in the FVC-ICAO dataset. The proposed technique that yielded the best results so far has a low computational cost and a simple implementation.

The remainder of this paper is structured as follows. Section 2 discusses related work on MTL and NAS. Sections 3 and 4 contain the experimental evaluations and discussion. Finally, Section 5 summarizes the results and future works.

2 RELATED WORK

MTL techniques may be divided into two categories: *Soft-Parameter Sharing* and *Hard-Parameter Sharing* (Vandenhende et al., 2021). In Hard-Parameter Sharing, a shared encoder branch is subdivided into task-specific branches that specialize in solving a single task. In Soft-Parameter Sharing, there is an algorithm to find where to share or branch within the network automatically and the task branches intersect at multiple points. In this work, we focus on hard-parameter sharing methods as they produce memory and computation efficient MTL networks (Zhang et al., 2022).

Some hard-parameter sharing MTL branches from a backbone model (Suteu and Guo, 2019; Leang et al., 2020) and from a single point in the network, similarly as we do in this research. A recent work (Zhang et al., 2022) proposes a tree-structured multi-task model recommender, which respects a user-defined computation budget. Previous works are based on task-relatedness calculation (Lu et al., 2017; Vandenhende et al., 2019) while other works, such as AdaShare (Sun et al., 2019) and AutoMTL (Zhang et al., 2022), learn a task-specific policy to select the layers that should be executed for a given task during the MTL network training.

Based on the successful results achieved by Zoph and Le (Zoph and Le, 2017) in NAS, the Efficient Neural Architecture Search (ENAS) (Pham et al., 2018) strategy applies the one-shot model strategy (Elsken et al., 2019). Similarly, Differentiable Architecture Search (DARTS) (Liu et al., 2018b) uses a one-shot model strategy, but performs the search using gradient descent algorithms. Other works such as (Xie et al., 2019) also uses RL, but the feedback mechanism is changed from fixed rewards, such as validation accuracy, to a generic loss calculated during training. Finally, (Liu et al., 2018a) used sequential model-based optimization (SMBO) to search.

3 MATERIALS AND METHODS

In this section, we present the methodology, dataset, evaluation metrics, architecture of each approach, and the idea behind them.

3.1 Methodology

Figure 2 summarizes the methodology. We used FVC-ICAO dataset with data augmentation (explained later), and experimented with five different network designs from distinct paradigms: Handcrafted STL, Handcrafted MTL and NAS. We started with STL as a baseline, and then investigated MTL with different setups and network designs. Finally, inspired by works found in the literature (Zoph et al., 2018; Pham et al., 2018), we proposed and evaluated a NAS approach. First, we designed a random approach for baseline purposes, and next, we implemented the Reinforcement Learning approach. The following sections provide more details.

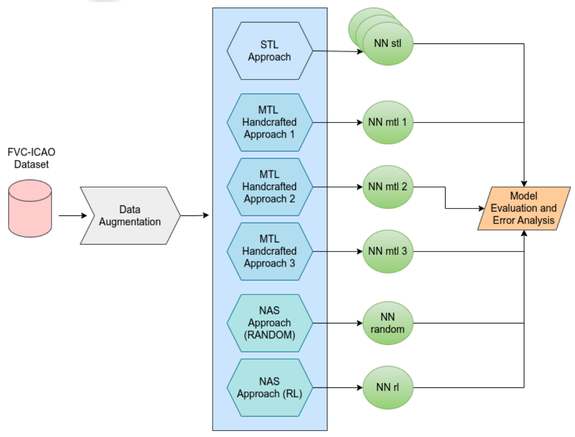

Figure 2: Scheme of the method, specifying the dataset and each approach proposed, which are evaluated with common metrics (accuracy and EER).

3.2 FVC-ICAO Dataset

The FVC-Ongoing competition (Ferrara et al., 2022) built the FVC-ICAO dataset as a reference for requisites compliance checking (Ferrara et al., 2012). Also, we included other *ad-hoc* images for increasing the number of available samples by ICAO requisite. The dataset was partitioned for training, validation and test (75%-15%-10%). The dataset has 5,865 images in total. We performed the following data augmentation for the FVC-ICAO dataset: horizontal flips, rotations, scale changes and intensity shear. All scripts and auxiliary material are available on a GitHub repository[1]. We used Tensorflow and Keras frameworks for training and for data augmentation.

3.3 Metrics

In this study, we computed the accuracy, which is defined by the Equation 1, where TP, TN, FP, and FN are the number of true positives, true negatives, false positives, and false negatives, respectively (Guido, 2017). We also used EER, a common metric for evaluating biometric systems performance, which is the error rate at a specific threshold *t* in which False Non-Match Rate (FNMR) and False Match Rate (FMR) are equal (Maltoni et al., 2009).

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

3.4 Architectural Setups

3.4.1 Single-Task Learning

Initially, we made experiments with STL, where each task was an ICAO requisite. Our STL method uses the same single-task architecture for all 23 tasks separately and employs transfer learning. Different base models were considered: VGG16 (Simonyan and Zisserman, 2015), VGG19 (Simonyan and Zisserman, 2015), MobileNetV2 (Sandler et al., 2018), Inception-V3 (Szegedy et al., 2016) and ResNet50-V2 (He et al., 2016). The best base model in our tests was VGG16. We have frozen the base model weights, as illustrated in Figure 3, and trained just the dense layers and the classification layer, so the new tasks could be learned. Each single-task network specializes in determining if an input image is compliant or non-compliant with a specific ICAO requisite. The experiments and results are discussed in Section 4.

[1]https://github.com/guilhermemg/nas_v1

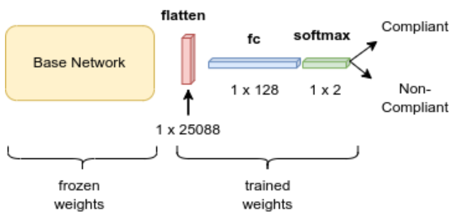

Figure 3: Single-Task Network architecture using Transfer Learning technique. *In red is the flatten layer, in blue the fully-connected layer, and in green the softmax activation.*

3.4.2 Multi-Task Learning

Next, we discuss each MTL architecture proposed.

Architecture HANDCRAFTED 1. The first MTL architecture tested was similar to the one used in the STL experiments. We used VGG16 as the base model, removing the original network output layer and freezing the trained weights. Also, new layers and an output layer were added, with 23 branches corresponding to each ICAO task. Figure 4 shows the general schema. A fully connected (FC) layer with 64 neurons followed by a FC layer with two neurons corresponding to the outputs of each task, compose each branch of the network. The activation function used in the output layer was softmax.

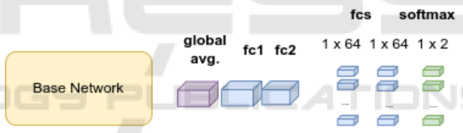

Figure 4: Multi-Task Learning *Handcrafted 1* architecture. *In purple is the global average pooling layer, in blue the fully-connected layers, and in green the softmax layer.*

Architecture HANDCRAFTED 2. Figure 5 presents the network. We maintained the general schema from architecture Handcrafted 1, but removed the shared branch and made all tasks branches linked directly to the Global Average Pooling layer. Also, we kept just one FC layer (1 x 64) for each task branch to test the learning potential of each branch with the minimum of FC layers possible. The number of FC layers is key for our research and is explained in more detail in Subsection 3.4.3 and in Section 4.

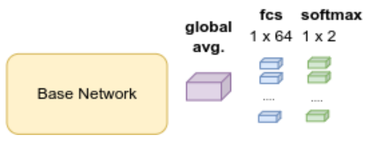

Figure 5: Multi-Task Learning *Handcrafted 2* architecture. *In purple is the global average pooling layer, in blue the fully-connected layer, and in green the softmax activation.*

Architecture HANDCRAFTED 3. Next, we grouped some ICAO requisites into common task branches with shared weights, testing the hypothesis that distinct tasks may benefit each other and increase the total gain while the network solves some tasks jointly. The ICAO tasks were hand-picked and grouped based on human discretion by analyzing their general characteristics, assuming they may share common features that the network could use to solve them with more efficacy. For example, if the network should analyze the face bottom half region for some tasks, then they should be in the same group. Figure 6 shows the proposed architecture. We also experimented increasing the number of FC layers per task branch. Thus, all 23 tasks branches have more FC layers than in the first and second studied MTL architectures, but the number of FC layers is fixed for each branch.

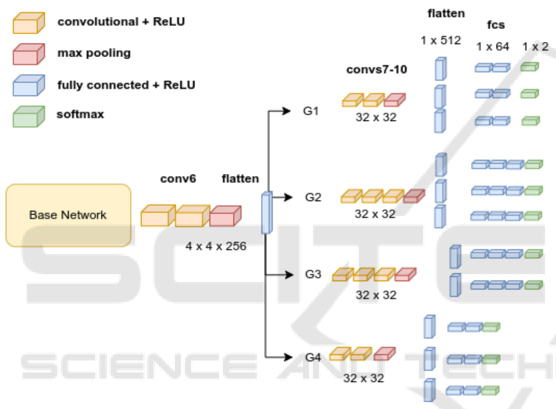

Figure 6: Multi-Task Learning *Handcrafted 3* architecture.

3.4.3 Neural Architecture Search

In order to reduce the search space and make the NAS experiments treatable, considering the resources available (just a single GPU), we restricted the problem to searching for the number of FC layers in each branch. Figure 7 shows the generic architecture on which the search is based. We maintain the base model and the general aspect of Handcrafted 3 MTL architecture. The neural architecture search consists in finding the values *n1*, *n2*, *n3*, and *n4*. These four values, which we call a ***config***, correspond to the four grouped tasks branches lengths, the sizes of the fully connected layers. Note Note that task group branches do not have convolutions and max pooling layers at this point to simplify the search and the overall implementation

The NAS process is depicted in Figure 8. The controller component (1) first selects a candidate architecture (2) and trains it in a validation set (3), then this candidate architecture is evaluated based on a chosen

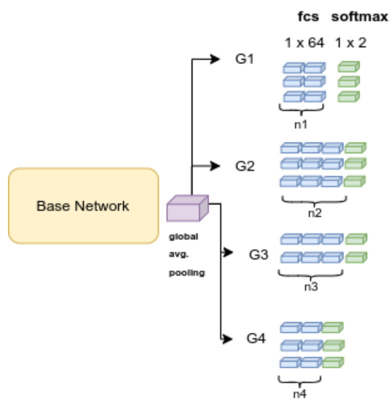

Figure 7: Neural Architecture Search generic architecture showing the searched parameters *n1*,*n2*,*n3* and *n4*. *In purple is the global average pooling layer, in blue the fully-connected layer, and in green the softmax activation.*

metric like accuracy or EER (4), and finally, the controller stores the result in memory (5). This process occurs for some iterations (also called *trials*), and the architecture with the best result found at the end is chosen as the search result.

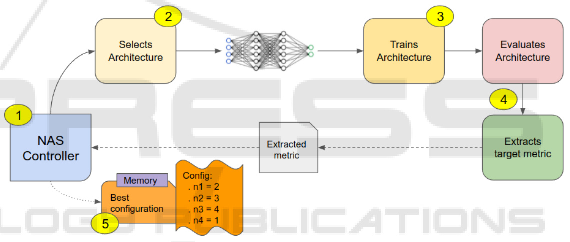

Figure 8: Neural Architecture Search basic process

We tried two base models, MobileNetV2 and VGG16, whose accuracies were similar. However, MobiletNetV2 inference time was much lower than VGG16, so, we selected MobiletNetV2 for all NAS experiments.

Random Search. We evaluated two different NAS approaches. The first one was a random search, referred to as *RANDOM*. We randomly chose from a predetermined integer interval (1-5) the *config* values and trained the random neural architecture. This interval was designed based on the available resources and also applies for the second search strategy.

Reinforcement Learning Search. We inspired our work on ENAS (Efficient Neural Architecture Search) (Pham et al., 2018). We train one LSTM network with 32 hidden cells as agent in the RL framework. The LSTM's input is the previous network configuration, which is used to propose a new architecture. The NAS process starts with a random *config* as the

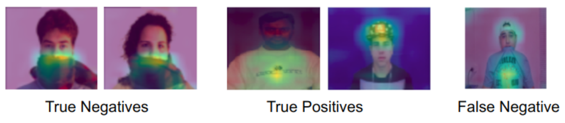

Figure 9: True negative, true positives, and false positive examples of Veil requisite with Grad-CAM heatmaps

first *trial*. Following training and evaluation, we calculate the mean accuracy of all tasks and record it in the NAS Controller memory with the respective *config*. Finally, the NAS Controller injects this last *config* and the obtained accuracy as achieved reward into the LSTM network and trains it for *m* epochs, which produces a new *config* of a new child network starting a new trial. The process repeats for a fixed number *N* of trials set in the NAS Controller instantiation. When the searching process ends, we recreate the best model architecture found, train it for 50 epochs, and evaluate it in the test set.

4 EXPERIMENTAL EVALUATION AND DISCUSSION

Table 1 shows the results obtained in the best resulting experiments for each approach, as presented next.

4.1 Single-Task Learning

The STL column in Table 1 shows the results obtained for each STL network training with ten epochs. We decided on ten epochs because the trainings converged with this number of epochs. All networks have approximately 3.2M trainable parameters from 17.9M parameters. In general, the results are not competitive. We did not evaluate the requisite Ink Mark in any experiment, since the random test set selected did not have instances of this requisite, so we could not calculate the metrics for it.

Error Analysis. Grad-Cam (Selvaraju et al., 2020) is a technique developed for aiding the explanation of CNN-based models decisions through visualizations produced on top of evaluated images. In Figure 9, the region that the network is paying attention to when taking the decision - compliant or not compliant - is highlighted in green and yellow. In contrast, the regions highlighted in red or violet are those the model is not paying attention to. For example, we can observe in Figure 9 that the network is looking for the region right below the person's nose when the person is wearing a veil. This pattern occurs in all true negative images of Veil requisite.

We also identified other patterns through Grad-CAM analysis. In the True Positives examples in Figure 9 the network pays attention to the person's shirt and headwear. Despite the network positive assertion (correctly classifying a compliant image) and high accuracy for this ICAO requisite, this can lead to failures in generalization.

Figure 9 also shows the single case where the network failed to identify that the person was not wearing a veil. We can check that the case is dubious since the person has the face partially occluded by the shirt.

The Grad-CAM analysis suggests that a MTL approach may be successful for the ICAO case: the fact that the network makes hits in one task while looking into regions of interest of other tasks reinforces the hypothesis that jointly learning the tasks may be beneficial.

4.2 Multi-Task Learning

In this section, we discuss the MTL approach results.

Architecture HANDCRAFTED 1. We performed three experiments with the Handcrafted 1 approach. In the first one (**Exp. I**), we trained the network by ten epochs and observed the evolution of training curves: *accuracy vs. epoch* and *loss vs. epoch* checking the training convergence. In the second one (**Exp. II**), we increased the number of trained epochs to 200 to see if the final accuracy would be higher with more training epochs, and again observed the training curves. We have not used early stopping since we would like to observe the whole training results towards all epochs. Lastly (**Exp. III**), we tried to fine-tune the base model for ten epochs, froze the base model weights again, and train the whole model for 200 epochs, so we could test the effect of base model fine-tuning for some epochs during training. In all experiments, we selected the best model based on the epoch with the highest validation accuracy. Table 1 summarizes the results.

It is possible to observe significant improvements through the experiments I to III with most ICAO requisites ending below 10% EER threshold: *Eyes Closed*, *Close*, *Flash Lenses*, *Light*, *Veil*, *Shadow Head* and *Hair Eyes*, and a group of ICAO tasks that were even better with a mean EER of less than 2%: *Hat*, *Dark Glasses*, *Washed Out*, *Red Eyes*. Comparatively to the STL approach, most requisites also had a better result in this MTL approach.

It is important to mention that the great change in the EER result for the Frames Heavy case - from 0.88% to 50% - was mainly due to the class unbalancing: we had just two samples of NOT COMPLAINT

Table 1: Mean EER, EER standard deviation, and Median EER on test set for STL and MTL best resulting experiments: MTL Handcrafted 1 - **Exp. III**, MTL Handcrafted 2 - **Exp. IV**, and MTL Handcrafted 3 - **Exp. III**.

ICAO Rq.	STL	MTL 1	MTL 2	MTL 3
Mouth	21.20%	6.0%	5.06%	**3.67%**
Rotation	27.94%	27.06%	18.27%	18.75%
L. Away	21.04%	11.46%	8.83%	11.96%
Eyes Closed	18.31%	2.04%	**1.98%**	3.75%
Close	3.32%	7.69%	4.02%	**0.00%**
Hat	1.40%	**0.98%**	2.88%	1.61%
Dark Glasses	1.81%	1.62%	0.20%	**0.00%**
Fr. Heavy	50.00%	50.0%	50.00%	**0.00%**
Fr. Eyes	15.58%	**3.74%**	5.08%	4.83%
Flash Lenses	11.76%	4.08%	4.70%	**3.86%**
Veil	2.38%	**2.38%**	2.94%	4.82%
Reflection	19.40%	15.32%	13.87%	13.17%
Light	14.63%	9.63%	**8.03%**	8.22%
Sh. Face	15.18%	17.96%	11.39%	12.13%
Sh. Head	9.18%	8.21%	9.72%	6.16%
Blurred	10.21%	12.39%	11.25%	9.67%
Skin Tone	21.13%	19.39%	16.64%	**0.00%**
Washed Out	1.06%	0.35%	**0.18%**	19.10%
Pixel.	31.79%	34.86%	22.99%	**0.00%**
Hair Eyes	12.94%	**2.68%**	4.73%	27.39%
Background	7.30%	21.54%	**3.87%**	4.89%
Red Eyes	14.40%	**1.77%**	1.98%	10.42%
Mean EER	15.09%	11.87%	9.48%	**7.47%**
EER sd.	11.54%	12.61%	10.91%	**7.32%**
Md. EER	14.52%	7.95%	5.07%	**4.86%**

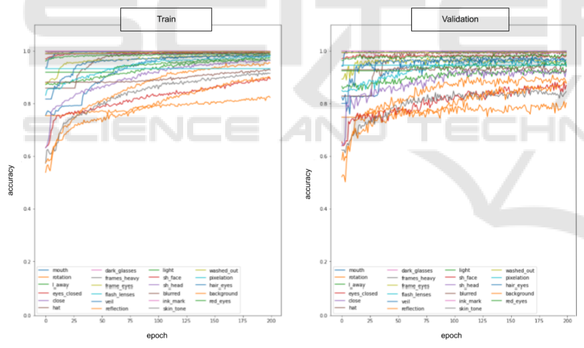

Figure 10: Training curves of MTL training - Experiment III - 200 epochs after 50 epochs of fine-tuning.

images (0.69%) for this requisite in a total of 288 test images. So, in cases like this, a small change in the output may cause great variations over the final EER. Consequently, we decided to use the *Median EER* in addition to the *Mean EER* as this statistic is less vulnerable to outliers.

Architecture HANDCRAFTED 2. We executed five experiments for the Handcrafted 2 approach: the first and second resulted from trainings with 10 and 200 epochs, respectively, similar to Experiments I and II of the Handcrafted 1 approach.

The last three experiments tested modifications in data augmentation. In Exp. III, we did no rotations in the images and changed the range of width and height shift from 0.2 to 0.1 simultaneously. In Exp. IV and V, we did no rotations in the images and trained the network for 50 and 200 epochs, respectively.

The hypothesis tested in these experiments are (i) the negative effect of rotation for data augmentation, that could be harming the Rotation (Roll, Pitch, Yaw) requisite; (ii) we also evaluated the shifting operation, whose value could be inadequate; (iii) we varied the number of training epochs checking the results for each requisite and for the general set of requisites. Table 1 shows the best achieved results in Experiment IV in terms of Median EER.

We can observe that Exp. IV had the best results in terms of Median EER (5.07%) compared to the other ones. The Rotation requisite had slight improvement and showed the best result so far, confirming the hypothesis that doing rotations in the images during data augmentation was prejudicial. Other requisites also presented the lower EER so far: *Eyes Closed*, *Light*, *Background*, and *Red Eyes*. Considering that, We decided not to make any rotations and use the same width and height shifts during data augmentation for the subsequent experiments.

Architecture HANDCRAFTED 3. In this last MTL approach, we did three experiments (I, II, and III) varying solely on the number of training epochs (10, 50, and 200, respectively), evaluating the training convergence and final Mean EER and Median EER. The results are available in Table 1, relative to Experiment III, which presented the best Median EER.

We can see that the requisites *Close*, *Dark Glasses*, *Frames Heavy*, *Skin Tone*, and *Pixelation* had 0% EER in the test set. Skin Tone and Pixelation were two difficult tasks, considering the high EER achieved by the other approaches (above 10% on average). In the next phases of this research, we will investigate these results more deeply.

4.3 Neural Architecture Search

In the NAS context, we analyzed different dimensions in our experiments: the *number of epochs* ($m \in [1,5]$) and the *number of trials* ($N \in [3,50]$). Table 2 show the results in the FVC-ICAO dataset for each different approach. In this case, we may note a difference between the proposed REINFORCE approach and the baseline RANDOM approach. The first one achieved 5.85% EER median as the best result, while the second one achieved 6.5% EER median. Curiously, increasing the number of epochs m generally did not improve the method efficacy in terms of median EER. Comparatively to MTL, NAS REINFORCE obtained a promising result, which is about just one percentual point below the 4.86% best MTL result (Handcrafted Approach 3).

Table 2: Results of Approaches 1 (RANDOM) and 2 (RL) in FVC-ICAO dataset in terms of EER Mean, EER standard deviation, and EER Median.

Ap.	*m*	*N*	EER Mean	EER sd.	EER Md
1	1	3	7.86%	7.2%	**6.5%**
1	1	50	7.72%	5.81%	7.69%
1	5	3	7.67%	6.79%	6.68%
1	5	50	8.55%	7.17%	9.2%
2	1	3	8.83%	7.52%	8.93%
2	1	50	7.73%	7.36%	**5.85%**
2	5	3	8.14%	6.79%	8.45%
2	5	50	8.31%	6.85%	7.83%

We intend to improve the search space with more operations - like skip-connections, concat, and splits - and different types of layers - such as 3x3 and 5x5 convolutions. We expect that by implementing these extensions to our method, we may achieve better results on the FVC-ICAO dataset, even surpassing MTL results and getting closer to the state-of-the-art on this dataset. Using a search space with these operations allows the proposition of an architecture such as Handcrafted MTL 3 by the NAS method.

4.4 Comparison with Literature

A direct comparison of our results with the ones available on the FVC-Ongoing competition is not yet possible, since we still need to make adjustments to the executable code of our model to meet the competition requirements in terms of size and execution time. However, considering two caveats (i) that FVC-Ongoing test set is different than ours and (ii) we could not evaluate the requisite Ink Mark, we decided to assess our method performance compared with some top solutions submitted to the competition platform. Note that we considered only solutions that evaluate all 23 ICAO requisites, so we can make a more fair comparison with our model. Table 3 shows the results of BioLab (Ferrara et al., 2012), BioTest (BioTest, 2017), and Biopass Face (Vsoft, 2017).

Table 3: EER of submitted solutions to FVC-Ongoing competition by independent developers, private companies, and academic institutions. Note, the platform uses its own test set, different of ours.

	BioLab	BioTest	Biopass Face
Mean EER	7.28%	9.89%	4.84%
EER std dev.	5.90%	9.05%	4.18%
Median EER	5.20%	5.10%	**3.10%**

Considering median and mean EER as reference metrics, our solution had competitive results. It is relevant to highlight that these competitors may have implemented solutions specifically designed for each one of these requisites. For example, an SVM for Dark Glasses, simple filters for Pixelation and Blur, neural networks for Veil, etc. In our solution, all the requisites are analyzed in a single neural network simultaneously.

In the future, we will also submit our model to the FVC-Ongoing platform, gather more precise results, and provide a deeper analysis on this topic relative to the FVC-ICAO dataset and requisites. Also, we will evaluate our method on other datasets like CelebA (Liu et al., 2015) and CIFAR-10 (Krizhevsky, 2009).

5 CONCLUDING REMARKS

This research ultimate goal is to develop a Neural Architecture Search (NAS) method with proven efficacy in ICAO requisites compliance checking and applicable in other contexts of Multi-Task Learning (MTL) such as MNIST, FASHION-MNIST, CIFAR-10, and Celeb-A. These datasets are commonly used in the evaluation of NAS methods. We initially proposed a method of NAS based on a previous work presented in the literature. The proposed method uses REINFORCE algorithm to train a Network Controller (LSTM) to find the best neural net architecture given a small set of parameters, such as the maximum size of a branch and the number of training epochs. The neural net found after the search is evaluated as the best to solve a given set of tasks simultaneously on a specific dataset. The final objective is that the implemented method of NAS can automatically group tasks and create branches inside the neural network architecture.

We evaluated our proposed approach of NAS initially based on FVC-ICAO dataset and compared the achieved results of NAS with results obtained with handcrafted STL and MTL methods and literature methods. The preliminary results of the NAS REINFORCE approach are competitive, with a 5.86% median EER in the FVC-ICAO dataset. The main conclusion of this paper is that even with a simple neural architecture search method, it is possible to achieve reasonable results close to human handcrafted architectures.

ACKNOWLEDGEMENTS

This study was financed in part by CNPq research agency (https://www.gov.br/cnpq) and by VSOFT company (https://www.vsoft.com.br/).

REFERENCES

BioTest (2017). Result of algorithm biotest 1.3.8 on ficv-1.0. https://biolab.csr.unibo.it/FvcOnGoing/UI/Form/AlgResult.aspx?algId=2787. Visited: Dec-2022.

Caruana, R. (1997). Multitask Learning. *Machine Learning*, 28(1):41–75.

Elsken, T., Metzen, J. H., and Hutter, F. (2019). Neural architecture search: A survey. *Journal of Machine Learning Research*, 20:1–21.

Ferrara, M., Franco, A., Maio, D., and Maltoni, D. (2012). Face image conformance to ISO/ICAO standards in machine readable travel documents. *IEEE Transactions on Information Forensics and Security*, 7:1204–1213.

Ferrara, M., Franco, A., Maio, D., and Maltoni, D. (2022). FVC-ongoing. benchmark area: Face image ISO compliance verification. https://biolab.csr.unibo.it/FVCOnGoing/UI/Form/BenchmarkAreas/BenchmarkAreaFICV.aspx. Visited: Dec-2022.

Guido, A. C. M. S. (2017). *Introduction to Machine Learning with Python*. O'Reilly, third edit edition.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Identity Mappings in Deep Residual Networks. *LNCS*, 9908:630–645.

ICAO (2015). Doc 9303 - Machine Readable Travel Documents - Part 1: Introduction - 7th Edition.

ISO (2017). ISO/IEC 19754-5 information technology — biometric data interchange formats — part 5: Face image data. https://www.iso.org/standard/50867.html. Visited: Dec-2022.

Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images. Technical report, MIT.

Leang, I., Sistu, G., Bürger, F., Bursuc, A., and Yogamani, S. (2020). Dynamic task weighting methods for multi-task networks in autonomous driving systems. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE.

Liu, C., Zoph, B., Neumann, M., Shlens, J., Hua, W., Li, L.-j., Fei-fei, L., Yuille, A., Huang, J., and Murphy, K. (2018a). Progressive Neural Architecture Search. *Proceedings of the 15th European Conference on Computer Vision*, pages 19–34.

Liu, H., Simonyan, K., and Yang, Y. (2018b). Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738.

Lu, Y., Kumar, A., Zhai, S., Cheng, Y., Javidi, T., and Feris, R. (2017). Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5334–5343.

Maltoni, D., Maio, D., Jain, A. K., and Parbhakar, S. (2009). *Handbook of Fingerprint Recognition*. Springer, 2nd edition.

Pham, H., Guan, M. Y., Zoph, B., Le, Q. V., and Dean, J. (2018). Efficient Neural Architecture Search via parameter Sharing. *35th International Conference on Machine Learning (ICML 2018)*, 9:6522–6531.

Ruder, S. (2017). An Overview of Multi-Task Learning in Deep Neural Networks. *arXiv*, pages 1–14.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 4510–4520.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2020). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision*, 128(2):336–359.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, pages 1–14.

Sun, X., Panda, R., and Feris, R. (2019). AdaShare: Learning what to share for efficient deep multi-task learning. *arXiv*, pages 1–19.

Suteu, M. and Guo, Y. (2019). Regularizing deep multi-task networks using orthogonal gradients. *arXiv preprint arXiv:1912.06844*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Vandenhende, S., Georgoulis, S., De Brabandere, B., and Van Gool, L. (2019). Branched multi-task networks: deciding what layers to share. *arXiv preprint arXiv:1904.02920*.

Vandenhende, S., Georgoulis, S., Van Gansbeke, W., Proesmans, M., Dai, D., and Van Gool, L. (2021). Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*.

Vsoft (2017). Result of algorithm biopass face 5.6 on ficv-1.0. https://biolab.csr.unibo.it/FvcOnGoing/UI/Form/AlgResult.aspx?algId=6336. [Visited Dec-2022].

Xie, S., Zheng, H., Liu, C., and Lin, L. (2019). SNAS: stochastic neural architecture search. In *International Conference on Learning Representations*, pages 1–17.

Zhang, L., Liu, X., and Guan, H. (2022). A tree-structured multi-task model recommender. *arXiv preprint arXiv:2203.05092*, pages 1–22.

Zhang, Y. and Yang, Q. (2021). A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, pages 5586–5609.

Zoph, B. and Le, Q. (2017). Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*.

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning Transferable Architectures for Scalable Image Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 8697–8710.