Abstract—In this paper, we deal with the automatic segmentation of guitars that appear in RGB images towards a better understanding of multimedia concert contents. We apply Convolutional Neural Networks (CNNs) between two classes: guitar and non-guitar, to detect the image pixels belonging to guitar regions. In this work, we compare three semantic segmentation methods in the literature: DenseNet, DeeplabV3+ and HRNet based encoder-decoder networks. These networks are trained and tested with a manually annotated dataset created for this purpose, which addresses musical concerts and live scenarios. We measure the variance of the results against lack of examples, class unbalancing and noise. The results of this research indicate that HRNet-based networks perform better in guitar segmentation, showing robustness while reducing false positive and false negative rates.

Index Terms—Guitar segmentation, Concert segmentation, Semantic segmentation, CNN

I. INTRODUCTION

In this work, we are interested in exploring how semantic segmentation networks help enhancing the objects segmentation in context-specific datasets. More specifically, in musical contexts, we are interested in segmenting guitars, which are considered an important source of information and the main problem of occlusions for the guitar players. If we detect the guitars, we are able to link the characters to their musical activity or reconstruct the character correctly in 3D for Virtual Reality applications. An efficient detection of guitars is necessary to obtain a better semantic understanding of concert images for high level applications.

Many researchers have developed methods devoted to semantic segmentation in RGB images. Results reported by Fully Convolutional Networks (FCN) on classification and later segmentation and detection tasks [1]–[3] led to a large variety of developed models. Objects segmentation requires the preservation of resolution-related information and extract scale-aware features. Among these variants, atrous convolution [4], encoder-decoder models [3], [5], [6] and depth-wise convolution [7] have emerged enhancing the performance of the first CNN architectures. Depending on the requested task, new strategies follow hybrid approaches in order to exploit the best characteristics of each method.

a) Encoder-Decoder: This kind of network has typically two phases: first, it reduces the feature maps to catch semantic information; then it recovers spatial information with up-sampling techniques. This approach has proven successful in segmentation [2], [3], [5], [6]. The Xception module, which modifies Inception V3 to strongly boost performance on large-scale datasets, is now used as main backbone in server environments [4].

b) Densely Connected CCNs.: They are considered a logical extension of Resnet, where each layer is directly connected to every other layer in a feed-forward fashion. This structure achieves a low number of trainable parameters exploiting the U-Net approach [5], [8], [9].

c) Recovering high-resolution features: Encoder architectures retain the low-resolution features from which we try to recover the higher ones later with the decoder. A recent work about High-Resolution Network (HRNet) [10], [11] recently demonstrated very good performances.

In this work we benchmark the performance in guitar segmentation of networks based on encoder-decoder models, which performs better with different context-based datasets. The different methods are:

- DeepLabv3+, with atrous convolution [4] using the adapted Xception backbone [7]
- Fully Convolutional Dense Networks [8], [9]
- High-Resolution Networks [10], [11]

II. METHOD

In order to validate the use of CNNs to obtain guitar segmentation from images, we compare and train some of the main semantic segmentation methods: DeepLabV3+ [4], DenseNet [8] and HRNet [12], applied in an encoder-decoder fashion. These implementations have been chosen according to the performance and efficiency achieved on the major public datasets.

A. Dataset

The importance of the type and quality of the data in machine learning has been widely recognized [13], [14],

This work is funded by the European Research Council (ERC) Advanced Grant Moments in Time in Immersive Virtual Environments (MoTIVE) number 742989, and partially funded by the National Science Center (Poland) by the grant UMO-2016/23/N/ST6/02076 and Supported by the Foundation for Polish Science (FNP)

978-1-7281-9617-6/20/$31.00 © 2020 IEEE
especially where supervised learning is concerned. Efficiency considerations over suboptimal datasets have been discussed to benchmark specific models [6]. Besides, the number of available examples remains a most-wanted requisite: in the last decade new very-large-scale datasets have emerged to push the state-of-the-art over each computer vision task. The most relevant are: PASCAL [15], Cityscapes [16], ADE20K [12], [17] and COCO [18].

In our study, we replicated the achievements of the methods implemented by respective authors; then, we addressed a general musical context, specializing on a binary semantic segmentation between guitar and non-guitar classes.

We have created a manually annotated Guitar Dataset in order to deal with guitar detection and segmentation. We group under this label electric guitars, acoustic guitars and basses. The main challenges provided by our dataset concern dealing with a small number of data and an high variance in geometrical and photographic features like exposure, blur and illumination.

1) Dataset creation: We used 2,200 RGB images with guitars in different contexts to train and test the chosen networks. For each image, we created a binary mask with 2 pixel-level classes: label 1 for guitar class and label 0 for non-guitar class. All the images were manually annotated with the segmentation of the guitars that appeared in the scene. Figure 1 first and second column shows some examples from the annotated dataset.

Since the number of images probably is barely enough to feed the network and get appreciable results, we decided to apply a data augmentation technique. A combination of several pre-processing algorithms were applied: aspect-ratio-aware resizing; padding and cropping up to the indicated size, with a tolerance of 30% of non-zero values across the mask; color standardization with 4 templates; binary color indexing (1 for guitar class and 0 for non-guitar class). Inputs with variable sizes are treated by Tensorflow or PyTorch to get proper crops for training. An upper limit was set over these sizes for validation purposes.

Therefore, from the original set of 2,200 images, we obtain an augmented dataset of 10,400 images and labels, which is divided in a training set of 8,200 examples and a validation set of 2,200 examples. TABLE I summarizes the dataset information.
B. Pretrained models

The chosen network architectures are devoted to different contexts, but they can be adapted to work with our guitar dataset via fine-tuning.

For DeeplabV3+, we used the pre-trained models that have proven to be reliable [4]. We adapt the weights to our own dataset via fine-tuning. The Initial checkpoints are the ones trained on PASCAL VOC Dataset, using the Xception backbone. Batch normalization layers were pretrained too and freezed to a limit computational complexity. We kept a learning rate of $10^{-4}$, with a polynomial decay factor of $0.1$. The batch size is fixed to 8.

For DenseNet, we utilized the implementation detailed in [8]. The model presented by the authors have been trained over the CamVid dataset [19], it has 2,288,560 parameters trained using stochastic gradient descent with a categorical cross-entropy loss function. The network ran for 20 epochs with learning rate of 0.005 and batch size of 4. Accuracy was initially measured with Dice coefficient, achieving a 86.56% value on the CamVid validation set during the training.

For HRNet, we used the structure defined in [12] which is provided also as a PyTorch implementation. Models are arranged as encoders plus decoders: we use HRNet as encoder and a unique convolution module as final decoder for feature upsampling. For this reason, while listing results we address the network as HRNet-C1. We finetuned the model previously pretrained on ADE20K dataset.

C. Training and validation

We launched our experiments on a server sharing 4 GPUs: a NVIDIA Quadro P6000 with 24GB memory, a NVIDIA Titan X with 24GB memory and two NVIDIA GeForce RTX with 11GB memory. CUDA version used is 10.0, which fulfills both TensorFlow and PyTorch requirements of our models.

In order to measure the performance, the mIoU (Mean Intersection Over Union) has been chosen, which evaluates the obtained segmented guitar regions with the ground truth segmentation. For binary (two classes) or multi-class segmentation, the mean IoU of the image is calculated by taking the IoU of each class and averaging them. This index is formulated as follows:

$$\text{IoU} = \frac{TP}{TP + FP + FN},$$

where TP is the number of True Positive pixel-level predictions, while FP and FN are the ones of False Positives and False Negatives respectively.

We benchmark the results according to the number of iteration steps used in the training process. We then observe the variation in mIoU and in time needed for the computation. All networks work evaluating epochs from the number of iterations desired and the amount of iterations to be performed to cover the whole dataset. Hence $n_{epochs} = \frac{n_{iter\_total}}{n_{iter\_epoch}}$.

III. Results

We have performed a complete evaluation for each of the three networks trained to obtain an automatic guitar segmentation from RGB images. Qualitative and quantitative evaluations have been made in order to test the performances of Deeplabv3+ [4], DenseNet [8] and HRNet [12].

The target number of iterations applied in the training process are both 40,000 (40K) and 80,000 (80K). We started our tests at 40K iterations since it is the number of iterations suggested by the author implementation of Deeplabv3+ [4] and
it has shown good results as a baseline on other datasets [18]. Once 40K iterations were examined, we decided to double the number of iterations to 80K in order to analyse the performance with a more exhaustive training process.

| Network      | Crop | Batch | mIoU (%) |
|--------------|------|-------|----------|
| DeeplabV3+   | 513  | 8     | 88.47    |
| FC-DenseNet103 | 256 | 4     | 54.22    |
| HRNet-C1     | 960 x 720 | 4 | 95.71    |

TABLE II
COMPARISON ON EVALUATION SETS APPLYING 40K TRAINING ITER.

Performances of all the networks with 40K iterations are shown in TABLE II. As we can observe, DeeplabV3+ and HRNet obtained a mIoU of 88.47% and 95.31% respectively, while DenseNet just reached a mIoU of 54.22%.

To test a more exhaustive training with 80K iterations with the same crop and batch size, we focused on both DeeplabV3+ and HRNet networks since they highly overcome the performance of DenseNet. These results are shown in TABLE III where we compare also the computational cost devoted to each training process. As we can see, HRNet obtains the best score with a mIoU of 95.35% while DeeplabV3+ scores a mIoU of 92.53%.

Results demonstrate that the context of training is crucial to enhance the segmentation results. Fully Convolutional DenseNet achieves state-of-the-art performances on street views, like other methods do [20], while on general purpose datasets, like ours, HRNet and DeeplabV3+ perform better. We can also observe that HRNet is faster at learning features in a good way: the mIoU is very high with 40K iterations on the dataset, while Deeplabv3+ needs 80K iterations to reach that value.

Figure 1 shows a qualitative comparison between ground truth and the outputs of all models in the basic experiment with 40K interations. As we can see, HRNet obtains the best score with a mIoU of 95.35% while DeeplabV3+ scores a mIoU of 92.53%.

In Figure 2 we can observe some results obtained by DeeplabV3+ and HRNet trained with 80K iterations. As we can see both networks perform correctly segmenting guitars. The difference between both results are mainly in some details of accuracy and precision where HRNet performs better, thus obtaining the best results.

| Network      | Train time | mIoU (%) |
|--------------|------------|----------|
| DeeplabV3+   | 41h        | 92.53    |
| HRNet-C1     | 53h        | 95.35    |

TABLE III
PERFORMANCES OF DEEPLABV3+ AND HRNET-C1 OVER 80K ITERATIONS.

IV. CONCLUSIONS

In this paper, we have presented a comparison between three reference semantic segmentation methods: DeeplabV3+ [4], DenseNet [8] and HRNet [10] networks, applied to the context of guitar segmentation in musical contexts. Applying 80,000 training steps, HRNet reaches the top value of 95.35%, while DeeplabV3+, scoring 92.53% is more efficient in terms of computational time. Guitar segmentation can be obtained by fine-tuning a pre-trained HRNet with the dataset presented in this paper, which is available in our web page.

REFERENCES

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, Nov 1998.
[2] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015, pp. 3431–3440.
[3] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” arXiv e-prints, p. arXiv:1409.1556, Sep 2014.
[4] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in ECCV, 2018.
[5] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” CoRR, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597
[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A deep convolutional encoder-decoder architecture for image segmentation,” CoRR, vol. abs/1511.00561, 2015. [Online]. Available: http://arxiv.org/abs/1511.00561
[7] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” CoRR, vol. abs/1610.02357, 2016. [Online]. Available: http://arxiv.org/abs/1610.02357
[8] S. Jégou, M. Drozdzal, D. Vázquez, A. Romero, and Y. Bengio, “The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation,” CoRR, vol. abs/1611.09326, 2016. [Online]. Available: http://arxiv.org/abs/1611.09326
[9] G. Huang, Z. Liu, and K. Q. Weinberger, “Densely connected convolutional networks,” CoRR, vol. abs/1608.06993, 2016. [Online]. Available: http://arxiv.org/abs/1608.06993
[10] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang, “High-resolution representations for labeling pixels and regions,” CoRR, vol. abs/1904.04514, 2019. [Online]. Available: http://arxiv.org/abs/1904.04514
[11] K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep high-resolution representation learning for human pose estimation,” CoRR, vol. abs/1902.09212, 2019. [Online]. Available: http://arxiv.org/abs/1902.09212
[12] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Scene parsing through ade20k dataset,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[13] Z. Qi, H. Wang, J. Li, and H. Gao, “Impacts of dirty data: and experimental evaluation,” CoRR, vol. abs/1803.06071, 2018. [Online]. Available: http://arxiv.org/abs/1803.06071
[14] V. Sessions and M. Valtorta, “The effects of data quality on machine learning algorithms,” 01 2006, pp. 485–498.
[15] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge: A retrospective,” International Journal of Computer Vision, vol. 111, no. 1, pp. 98–136, Jan. 2015.
[16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, s. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[17] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Semantic understanding of scenes through the ade20k dataset,” arXiv preprint arXiv:1608.05442, 2016.
[18] T. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick, “Microsoft COCO: common objects in context,” CoRR, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312
[19] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, “Segmentation and recognition using structure from motion point clouds,” in ECCV (1), 2008, pp. 44–57.
[20] Y. Yuan, X. Chen, and J. Wang, “Object-Contextual Representations for Semantic Segmentation,” arXiv e-prints, p. arXiv:1909.11065, Sep 2019.