Functional Specialization between Words and Music in Convolutional Neural Networks

Amin Heydarshahi AMIN.HEYDARSHAHI@FAU.DE
Chair of Visual Computing
Friedrich-Alexander-Universität Erlangen-Nürnberg
Erlangen, Germany

Supervisors: Prof. Dr. Bernhard Egger, Prof. Dr. Andreas Kist, Dr. Katharina Dobs

Abstract

How and why functional specialization occurs in the human brain has been the topic of extensive research. Recent advances in computer vision and deep learning have enabled the use of convolutional neural networks (CNNs) to provide deeper insights into this aspect of the brain. In this work we measure the extent of task segregation in CNNs on two tasks that are segregated in the brain, namely, music genre classification and spoken word recognition. Specifically, we explore the impact of different network architectures and batching techniques on the selectivity of the networks.

Keywords: Functional Specialization, Task Segregation, Convolutional Neural Networks

1. Introduction

Functional selectivity in the brain has been a topic of extensive research in neuroscience. Proving a brain region is specialized for a specific function is in and of itself a difficult challenge. Fusiform Face Area (FFA) is an example of such regions (Kanwisher and Yovel, 2006).

One interesting question is why specific functions—such as faces, objects, music, and spoken words—are selected by specific regions of the brain. Dobs et al. (2022) suggest a computational reason behind it. To analyze their hypothesis, they train three types of Convolutional Neural Networks: One trained only on faces, one trained only on objects and one dual-task network trained on both faces and objects. They conduct two experiments where they:

- Compare the performance of the dual-task network with single networks. They show that while dual-task networks can perform both tasks with the same accuracy as each of the single networks, features of one task are not suitable for learning the other task.
- Identify 20% of the most important CNN filters for one task, lesion them (i.e. set them to zero), and calculate the performance drop on the other task. This gives a measure of task segregation in the neural network.

By observing a significant segregation between the two tasks, they conclude a computational structure in the brain could explain the reason behind functional selectivity.

In this project, we aim to investigate the degree of task segregation between two other tasks in CNNs, namely, music genre classification and spoken work recognition. Building up on the work of Kell et al. (2018), we do the lesioning experiments from Dobs et al. (2022) on CNNs trained on music and word data. Furthermore, we explore the impact of different network architectures (e.g. AlexNet vs. VGGNet) as well as different data batching techniques (mixed- vs. non-mixed batches).

2. Methods

In this section, we specify the task definitions, dataset creation process, and the experiments.

The tasks include:

- **Music genre classification:** Classification task between 61 different genres.
- **Spoken word recognition:** Classification between 467 words classes

For both tasks, we define the objective function to be cross-entropy loss between the correct label and the predicted probabilities from the network.

2.1 Dataset

For the music genre classification task, we started with the MTG-Jamendo Dataset (Bogdanov et al., 2019) which includes 55,000 audio tracks with 195 tags. 87 out of these 195 were genre tags. We took the top 61 classes which contained more than 400 audio tracks for each genre. The resulting dataset was highly unbalanced, some classes had more than 15,000 audio samples while some had 400. To overcome this imbalance, we cropped random 2-second clips of the tracks from undersampled classes. As a result, each class contained 15,000 2-second audio samples after this step. Hence the total dataset count became 915,000. Finally, we extracted Mel-Spectrogram features of size (128×128) from the audio waveform and used it as the input of the networks.

As for the words dataset, we took a similar approach: we started with Common Voice Dataset (Ardila et al., 2020) which contains more than 1 million different recorded sentences. To keep the consistency in data format and sample count, we filtered the audio tracks to the sentences including the words that were used in Kell et al. (2018). We then removed the classes with less than 400 samples, which yielded a total of 467 words. The next step was to crop the audio tracks to 2-second clips around the target word. The cropping process consisted of finding the word position in the sentence text and estimating the timestamp where the word occurred based on it. Of course, this introduces some error, but a sample testing showed that more than 95% of the words would happen in the selected 2-second range. Similar to the music dataset, we resized the final audio clips to (128×128) Mel-Spectrogram features. All of the Mel-Spectrogram features were normalized to have a mean of 0 and standard deviation of 1 during training.
More details on the generation and parameters of the Mel-Spectrogram features can be found in Appendix 4

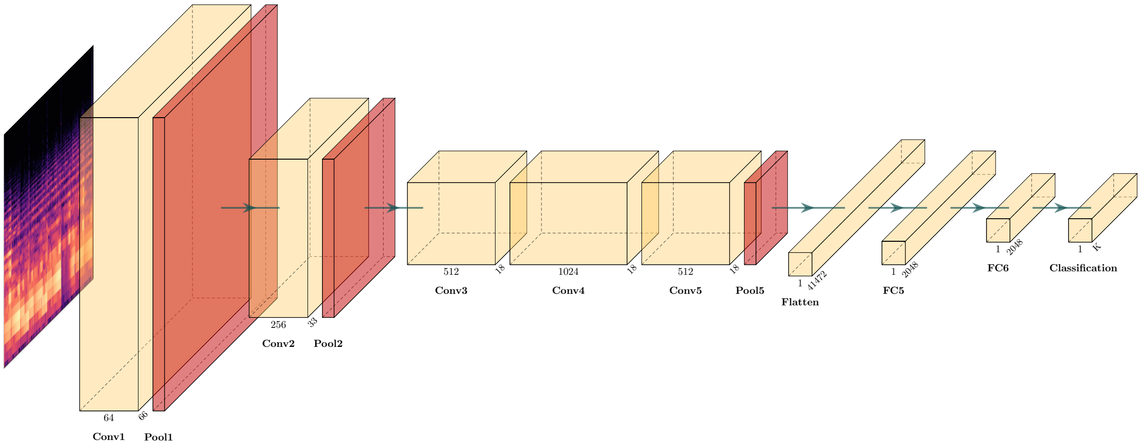


Figure 1: General structure in our AlexNet CNN. The output size K is set to 467 for word, 61 for music, and 528 for the dual task.

2.2 Networks

We did our experiments on two different CNN architectures:

- AlexNet: Inspired by Krizhevsky et al. (2012), this network consists of 5 convolutional layers followed by 2 fully-connected layers and a final classification layer. Figure 1 demonstrates this architecture[1].

- VGG-16: First introduced in Simonyan and Zisserman (2014), VGG-16 includes 13 convolutional layers, 2 fully-connected layers and a classification layer.

For the dual-task networks, output size is the aggregation of genre and word classes, which is $61 + 467 = 528$.
More network visualizations and full training hyperparameters can be found in Appendix 4

2.3 Experiments

We conducted three experiments:

2.3.1 Single vs. Dual-Task Networks

During this phase, we trained one word CNN, one music CNN, and one dual-tasked CNN and computed their top-5 test accuracy. Our main goal is to answer the question: Will the dual-task networks perform both tasks as well as single-task networks?

2.3.2 Mixed- vs. Non-Mixed Batches

We also investigate the effect of batching method on the dual-task networks. Specifically, we find out how the training batches are created and fed to the CNNs impacts the performance of the network. To that end, two different approaches for batching:

[1] The architecture is visualized using (Haris Iqbal, 2020)

- **Non-mixed batches:** with this approach, batches are created in turns based on the task. That is, we give one music batch as input to the network, compute the cross-entropy loss, update the weights, then create a word batch and repeat the process.

- **Mixed batches:** in contrast, we can have a batch consisting of both music and word samples. With this approach, the network sees both types of data in each learning iteration.

2.3.3 Lesioning Experiments

One way to measure the level of functional segregation is to lesion filters that are responsible for one task and observe the behavior of the network on the other task. Our goal with this experiment is to find out if we can see a selective impairment of the specialized function as is the case with the human brain.

More concretely, after the dual-task networks are trained, we examine the last convolutional layer of the networks (conv-5 in AlexNet and conv-13 in VGG-16). For each task, we disable the filters one by one and run the lesioned network on both data types (music and word), computing performance drop ratio as follows:

For a dataset X of size N and a filter f of the dual-task network—in this case, a melspec dataset matrix of $(N \times 128 \times 128)$—let $acc(X)$ be the mean accuracy of the network on the dataset and $acc_f(X)$ be the accuracy of the network when f is lesioned (set to zero). The drop ratio for this filter is computed as:

$$drop_f(X) = \frac{acc(X) - acc_f(X)}{acc(X)} \tag{1}$$

Our lesioning experiment then will have these steps:

1. Let F be the set of filters that are to be lesioned and X_w and X_m be the word and music data, respectively.

2. Compute $drop_f(X_w)$ and $drop_f(X_m)$ for all $f \in F$.

3. Sort the drop ratios and select the top-20% filters for each task.

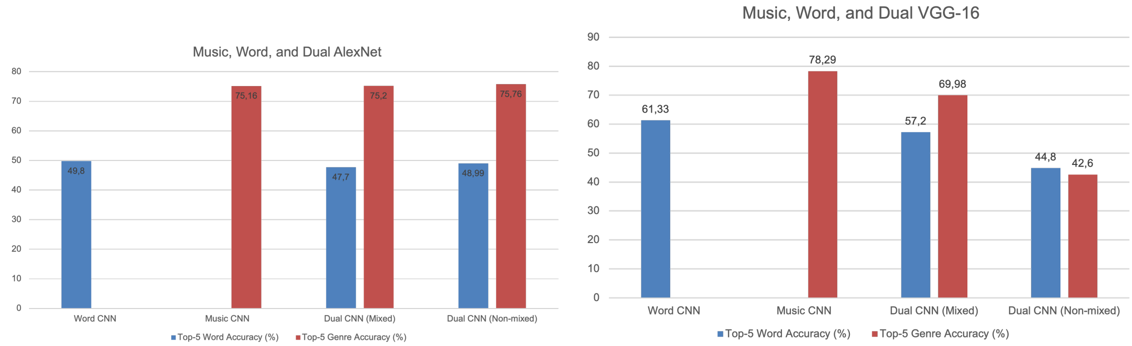
4. Lesion all of the top-20% filters for each task, evaluate the lesioned network on both tasks.

If lesioning top-20% of one task does not affect the performance of the other task, it suggests a high segregation in the CNN, while simultaneous performance drops will signal that filters have learned features that are common between the two tasks.

3. Results

Dual-task AlexNet networks can perform on par with single networks

After training all three types of network (word, music, and dual-task), we noticed that the dual-task networks can perform the same as single networks on both tasks in terms of top-5 accuracy (Figure 2a). We observed that with non-mixed batch training, the network overfit more often than it does with mixed-batch training.



(a) (b)

Figure 2: Comparison of top-5 accuracy of single- and dual-task networks with mixed and non-mixed training. (a) AlexNet top-5 accuracies. Both mixed and non-mixed training can perform both tasks as well as single-task networks. (b) VGG-16 top-5 accuracies. While mixed training can perform close to single-task networks, non-mixed training results in worse performance.

While this was the case for both mixed- and non-mixed batch trainings in AlexNet, the results are a little different.

Dual-task VGG-16 performs on par with single networks only with mixed-batch training

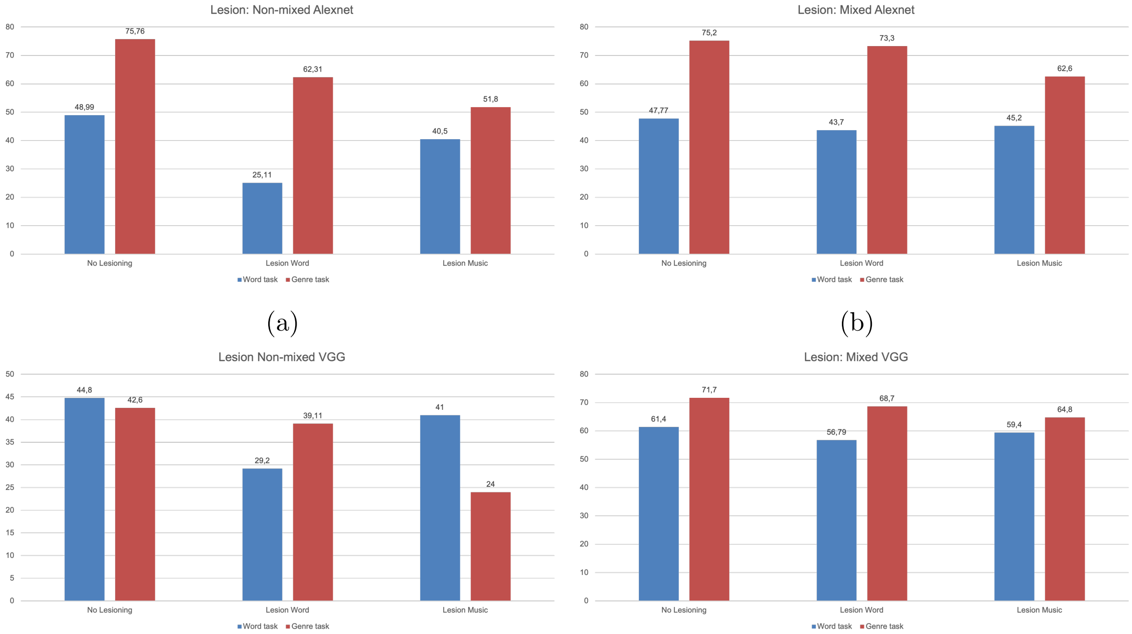
With VGG-16, we could not make the non-mixed models converge to the same accuracies as the single-task networks. In contrast, mixed-batch training almost always resulted in the same performance (Figure 2b). This can suggest that mixed-batch training is more robust to overfitting.

Lesioning shows more segregation in non-mixed batch networks than mixed-batch network As shown in figure 3, we can see that the lesioning of each task drops the performance of that task significantly while being indifferent to the other task in the non-mixed networks. With the mixed-batch CNNs, this effect is not as significant. This can mean that with non-mixed training, the network shows more segregation between the tasks.

4. Discussion

In this study, we explored task segregation between music and word in convolutional neural networks and examined the impact of two implementational approaches, namely architecture and batching, on the training and lesioning of CNNs. While this shed some light on the role these implementational details play, there are still myriad moving parts in the training process that can be explored further.

Some future directions include:



(a)

(b)

(c)

(d)

Figure 3: Lesioning experiments on AlexNet. (a) Lesioning on non-mixed batches suggests that lesioning top-20% filters of one task significantly drops the top-5 accuracy for the same task while having an insignificant drop in the other task. (b) With the mixed-batch model, we ovserve a much more slight amount of the lesioning effect. (c), (d) The same pattern can be observed in VGG-16 models.

- **Ablation studies:** to have a more complete picture of the lesioning experiments, we need to ablate various factors. This includes the tasks itself; an experiment is needed where we do the same lesioning experiment on two random subsets of music and word classes as done in Dobs et al. (2022).

- **Batch normalization:** we used batch normalization layers in both network architectures that we used. One can argue that this negates the effect of lesioning. While we conducted partial experiments without batch normalization layers—and it yielded more or less the same result—a more complete future experiment can bring to light more interesting results.

- **Lesioning method:** the lesioning done in Dobs et al. (2022) goes beyond our method by doing a greedy search on different subsets of the filters to find the top-20% most contributing filters for each task. This means we cannot directly compare our results with their lesioning of faces and objects.

- **Datasets:** using data different from our datasets can result in different and possibly interesting outcomes. This can be also the topic of a future experiment.

A major challenge of this project was the lack of a large-enough open dataset with spoken words; hence the most time-consuming part of this project was the generation of word and music datasets. Ultimately, the compiled datasets can facilitate any future work in the music and word research.

We hope that this project can be a starting point for further studies that can expose more aspects of learning not only in CNNs, but also in the human brain.

Acknowledgments

This project is conducted by Amin Heydarshahi as the 10-ECTS Project II of Artificial Intelligence M.Sc. curriculum at FAU Erlangen-Nürnberg. I would like to acknowledge and appreciate the support of my supervisors Prof. Dr. Bernhard Egger, Prof. Dr. Andreas Kist, and Dr. Katharina Dobs who provided me with the guidance for the experiments.

References

R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber. Common voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 4211–4215, 2020.

Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra. The mtg-jamendo dataset for automatic music tagging. In *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML 2019)*, Long Beach, CA, United States, 2019. URL `http://hdl.handle.net/10230/42015`.

Katharina Dobs, Julio Martinez, Alexander J. E. Kell, and Nancy Kanwisher. Brain-like functional specialization emerges spontaneously in deep neural networks. *Science Advances*, 8(11):eabl8913, 2022. doi: 10.1126/sciadv.abl8913. URL `https://www.science.org/doi/abs/10.1126/sciadv.abl8913`.

Pedro Diamel Marrero Fernández Haris Iqbal. Plotneuralnet. `https://github.com/HarisIqbal88/PlotNeuralNet/`, 2020.

Nancy Kanwisher and Galit Yovel. The fusiform face area: a cortical region specialized for the perception of faces. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 361(1476):2109–2128, 2006.

Alexander JE Kell, Daniel LK Yamins, Erica N Shook, Sam V Norman-Haignere, and Josh H McDermott. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630–644, 2018.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL `https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf`.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Appendix A. Specifications of Mel-Spectrogram Features

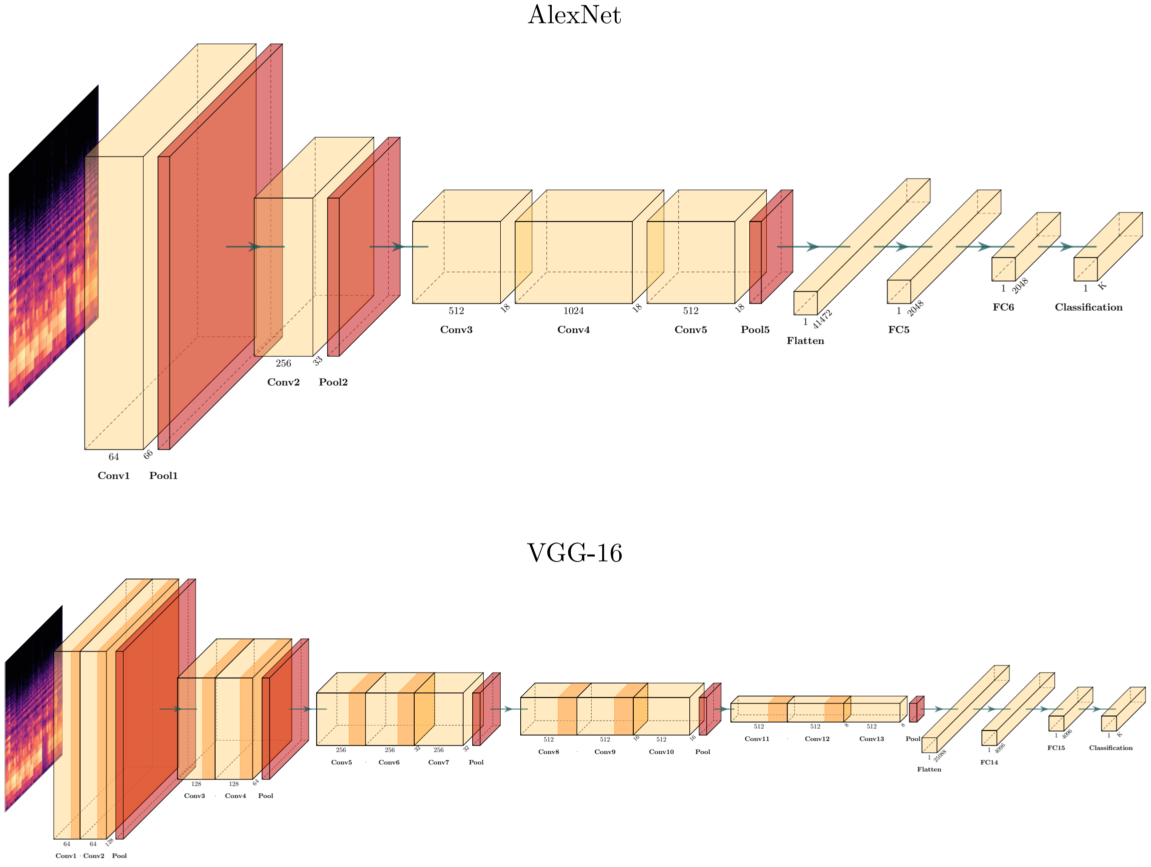
Here we show the important parameters that we used to generate both word and music Mel-Spectrogram features:

Sample Rate	**Window Size**	**Hop Size**	**Number of Bands**	**Output Size**	**Resized Output Size**
12000	512	256	96	(96, 95)	(128, 128)

Mel-Spectrogram important attributes and their values.

Appendix B. AlexNet and VGG Architectures and Hyperparameters

All of the models were trained on PyTorch and on V100 and A100 GPUs. The training of single-task networks was done in 10-15 hours for 10 epochs. The dual-task networks took up to 24 hours for 10 epochs of training.
Here we look at the two architectures, i.e., AlexNet and VGG-16:



AlexNet

VGG-16

Visualization of CNN architectures.

Finally, here are the hyperparameters we used for training AlexNet and VGG-16:

	Total Parameters	**Loss Function**	**Optimizer**	**Learning Rate**	**Weight Decay**	**Batch Size**
AlexNet	101,057,744	Cross-Entropy	Adam	5e-4 → 1e-5	5e-5	64
VGG-16	136,446,416	Cross-Entropy	Adam	1e-4 → 1e-6	5e-4	64

CNN training hyperparameters.