Mammalian Brain Classification

Idan Ludmir
ID - 205736887
Email - idanludmir@mail.tau.ac.il

Numerous studies have been performed using Deep Learning (DL) in brain MRI. However, there were no studies performed to research the differences in brain connectivity between different species using DL. This body of work seeks to solve the classification problem of different mammalian brains, taking one step closer towards obtaining an objective distance metric. Working on a partial set of 52 images taken from the MaMI dataset, which contains MRI scans of 123 species, I compare three training frameworks to classify between different taxonomic families. SupCon loss combined with transfer learning has been shown to obtain the best results on the dataset, with 8% increase in ACC comparing to the base model. The model achieved 78.41% total ACC on a set with 5 mammalian families, and 72.09% on a set with 12 families. These finding can assist in building a distance metric for mammalian brains, which can be compared to current taxonomy based on genetics and physical attributes. Furthermore, future work can involve training a model using weighted supervised contrastive learning, such that the embedding space will represent the actual evolutionary taxonomy.

I. INTRODUCTION

The nervous system in all vertebrate and most invertebrate animals is being controlled by the brain. This organ can contain around 160 billion cells for humans [1]. Around half of which are called neurons, and the other half is called glial cells [2]. Neurons communicate with one another through junctions called synapses, which are located at the ends of axons. The axons carry trains of electrical pulses called action potentials, which are generated in the cell's body (or Soma) and finish in the synapse. After arriving at the synapse the signal is then being transmitted to the adjacent cell's dendrites by extruding chemicals known as neurotransmitters [3].

These fairly simple units can connect to form much more elaborate networks. Those networks are being studied in a branch of neuroscience called "Connectomics" [4], which seeks to analyze the connectome; the map of neural connections in the brain, out of the realization that the structure and function of the brain are linked [5].

A common tool to analyze and extract brain imagery in order to study it's connections and therefore it's function is Magnetic Resonance Imaging (MRI) [6]. It is widely used in hospitals and clinics, as well as for research purposes. MRI images can be stacked to create a 3D image of the whole scanned organ. Compared to Computed Tomography (CT) it provides better contrast between grey matter and white matter. White matter refers to the axons of the neurons in the brain, because they are covered in a white substance called Myelin. Grey matter refers to the neuronal cell bodies, synapses and other non-neuronal cells in the brain.

This body of work focuses on two main MRI sequences: T1-weighted and T2-weighted. In T1-weighted the grey matter is viewed darker in comparison to the white matter, while in T2-weighted images the white matter is darker [7].

Numerous studies have been performed using Deep Learning (DL) in MRI. From image reconstruction [8] to segmentation [9], DL in MRI has already been used for diagnosis and prediction of various diseases [10][11]. There have also been some studies that use DL to study the brains of non-human subjects, for example rats [12]. However, there were no studies performed to research the differences between different species using DL. This body of work is based on the work of Yaniv Assaf et al [13], which gathered MRI images of 123 mammalian species in order to study their connectomes.

In evolutionary terms, mammals are animals constituting the class Mammalia, of clade Synapsida. They evolved from basal amniotes around 320 million years ago, where they diverged from sauropsids - the group that includes modern reptiles and birds [14]. Studying the similarities between mammalian species and the way their brains developed can assist in understanding the underlying mechanisms in the human brain. In this study I seek to show that DL can also be used as a tool to investigate those similarities, by the classification of MRI images to different mammalian families.

FIG. 1: Top 5 represented mammalian families in the dataset. Right to left, top to bottom - Bovidae, Cervidae, Canidae, Felidae and Mustelidae.

II. DATA

A. Sources

Yossi Yovel and Yaniv Assaf provided with a subset of the Mammalian MRI dataset (MaMI) [15], that includes 52 T2-weighted 3D images of 12 different mammalian families. The full list can be found in Appendix B. The dataset was unbalanced - see FIG. 2, so the 5 families with the highest amount of samples were chosen as the dataset for the initial experiments. Those families were: Bovidae, Cervidae, Canidae, Felidae and Mustelidae (see FIG. 1).

Due to the small dataset at hand (only tens of samples), I used a pretrained model (link), trained on Big Healthy Brains (BHB-10K) dataset [16], an aggregation of 13 publicly available datasets of 3D T1 MRI scans of healthy controls (HC) acquired on more than 70 different scanners and comprising $N = 10^4$ samples. Since the MaMI dataset includes T2 weighted images, I also ran a pre-training of the network on two datasets, taken from OpenNeuro, which contain T2-weighted images. One is the NIMH intramural healthy volunteer dataset [17][18], consisting of 157 human participants. The other is MultiRat_rest [19](GitHub link), consisting of 646 rat participants, later manually curated to reach a final amount of 192 images.

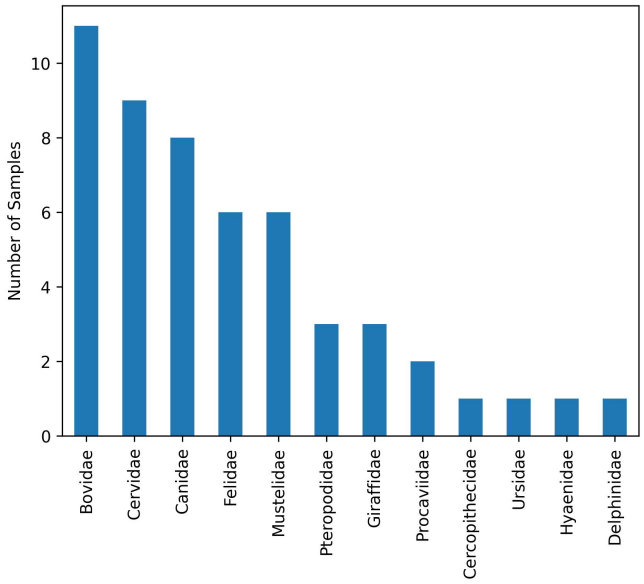


FIG. 2: Histogram of available mammalian families in the researched dataset.

B. Data preprocessing

Due to the fact that the MaMI dataset contains ex vivo MRI scans, the images taken from NIMH dataset were processed with FSL's [20] Brain Extraction Tool [21] to remove the skull surface and other non brain tissue from the images. The rat images taken from MultiRat_rest were processed using RBET [22], a modified version of BET that can process rodent brains.

To maintain consistency between all inputs, all of the images were resampled using FSLEyes [23], to an image size of $121 \times 145 \times 121$. Then, to increase the number of samples in the datasets, to minimize the number of features and to minimize memory footprint, 20 random crops in the size $80 \times 80 \times 32$ of each image were taken. That resulted in a pretraining dataset (consisting of the human and rat images) of 6900 images, and images from the MaMI dataset in a total of 800 images.

Examples for the preprocessing can be seen in FIG. 3.

(a)

(b)

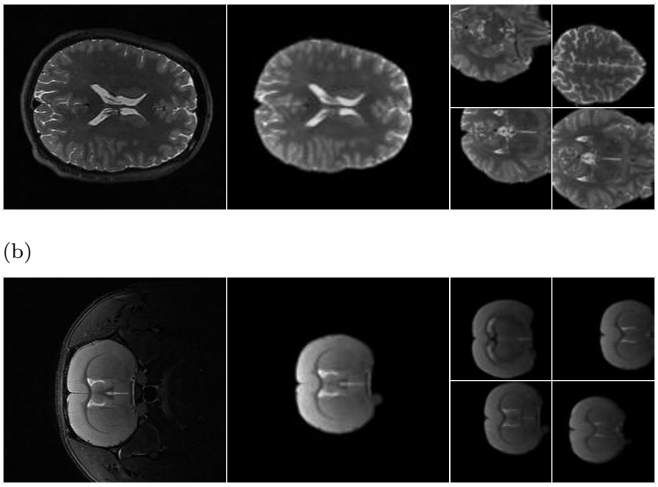


FIG. 3: Examples for the data preprocessing. From left to right - original, post brain extraction and resampling, final cropped augmented images. The crop slices are taken from the same index in every 3D image, to show that different areas in the brain are being sampled. (a) - Data from NIMH dataset. (b) - Data from MultiRat_rest.

III. METHODS

A. Contrastive Learning

Self-supervised learning can be regarded as an intermediate form between supervised and unsupervised learning. It is a method that does not rely on carefully curated and labeled datasets, which aids to its popularity today.

Contrastive learning is a form of self-supervised learning, where samples of the same data point are encouraged to have more similar representations compared to samples of different input points [24]. That way clusters of points belonging to the same class are pulled to-

gether in embedding space, while simultaneously pushing apart clusters of samples from different classes. Non-contrastive learning, in comparison, uses only inputs from the same label for the training. Samples from the same data point are called positive samples, and samples generated from different data points are called negative samples.

Another more recent form of contrastive learning is supervised contrastive learning [25]. Which extends self-supervised contrastive learning to a fully-supervised setting. This method leverages both the labels of the data and the improvements for the embedding space distribution that contrastive learning yields, such that it out performed other fully-supervised models.

Next, two frameworks that perform contrastive learning will be discussed, SimCLR - which does not require labels, and SupCon - which uses labeled data.

1. *SimCLR*

A simple framework for contrastive learning of visual representation, which learns representations by maximizing agreement between differently augmented views of the same data example [26]. That way there is no need for labeled data since augmentations from the same example are considered positive, while augmentations from different examples are negative. The contrastive loss function that is being used in this framework is NT-Xent (the normalized temperature-scaled cross entropy loss), and it is defined as:

$$\mathcal{L}^{self} = -\sum_{i=1}^{2N} log \frac{exp(z_i \cdot z_j/\tau)}{\sum\limits_{k \neq i}^{2N} exp(z_i \cdot z_k/\tau)} \tag{1}$$

Where i is the anchor index, j is the index of the positive pair, $2N$ is the number of augmented samples, τ is a scalar temperature parameter and z_x is the projection of the hidden representation.

The equation takes the form of a positive pair at the numerator, and $2N-1$ positive and negative pairs at the denominator. Note that for each anchor i, there is only 1 positive pair - since there are no labels, even if there are more samples that are part of the same class, the algorithm can not take them into account.

2. *SupCon*

Supervised contrastive learning loss takes into account the labels of the data, therefore generalizing EQ. 1 to an arbitrary number of positive pairs, and results in the following formula:

$$\mathcal{L}^{sup} = -\sum_{i=1}^{2N} \frac{1}{|P(i)|} \sum_{j \in P(i)} log \frac{exp(z_i \cdot z_j/\tau)}{\sum\limits_{k \neq i}^{2N} exp(z_i \cdot z_k/\tau)} \tag{2}$$

Where $P(i)$ is the set of indices of all positives in the batch besides the anchor i. $|P(i)|$ is the cardinality of the set.

B. Model architecture

The chosen architecture was Dense Convolutional Network (DenseNet) which, in comparison to a traditional CNN, contains connections between each layer to every other layer in a feed-forward fashion [27]. It has been wildly used for image classification tasks, also in the medical domain [28]. I used a modified version of DenseNet121 to support 3D images [16] - all kernels used in convolution, batch normalization and pooling layers were converted to 3d kernels.

Furthermore, the model was used in two configuration modes: an *encoder* configuration, and a *classifier* configuration. The encoder was used to pretrain the model using contrastive learning, while the classifier was used to fine tune the model on the MaMI data. The encoder configuration included a head projection layer after the hidden representation layer, that generated an output array of 128 cells. The classifier configuration replaces the head projection layer with another fully connected layer followed by a softmax layer, that outputs an array the size of the number of classes. A detailed representation can be seen in FIG. 4.

IV. EXPERIMENTS

A. Base Model

In order to test the progress, a base model has been trained to compare the rest of the results to. This model uses a simple cross entropy loss and no pretraining whatsoever. Setting the network to the *classifier* mode, it has been trained on 67% of the data and tested on the rest 33%. The train and test sets were saved for later use in the other models for consistency. This network is expected to overfit, or at least not generalize well since it has been trained on a small dataset.

Using Adam optimizer, learning rate was set to $1e{-}4$ and weight decay to $5e{-}5$. The network has been trained with a batch size of 10 for a 100 epochs.

Observing the loss values for each epoch (see Appendix C), the validation loss has been at its lowest point on epoch number 84. Therefore, the model from that epoch was chosen to be used as the base model.

	Bovidae	Cervidae	Canidae	Felidae	Mustelidae	Mean	Total
base	73.61	**67.74**	79.59	51.28	80.95	70.634	71.21
pretrained	84.72	53.23	**87.76**	64.10	80.95	74.16	74.24
pretrained + SimCLR	75.00	**67.74**	79.59	51.28	73.81	69.484	70.45
pretrained + SimCLR BS32	66.67	69.35	65.31	64.10	**85.71**	70.23	69.70
pretrained + SupCon	**86.11**	66.13	81.63	**71.79**	**85.71**	**78.274**	**78.41**

TABLE I: Per-class accuracy for each tested model, together with the mean-per-class accuracy and total accuracy. Highest scores for each category are highlighted in bold. *base* is the base model, *pretrained* is the model pretrained on BHB10K and OpenNeuro data, *pretrained + SimCLR / BS32* are the models created using SimCLR on the MaMI data with batch size 10 and 32 respectively, *pretrained + SupCon* is the model created using SupCon on the MaMI data.

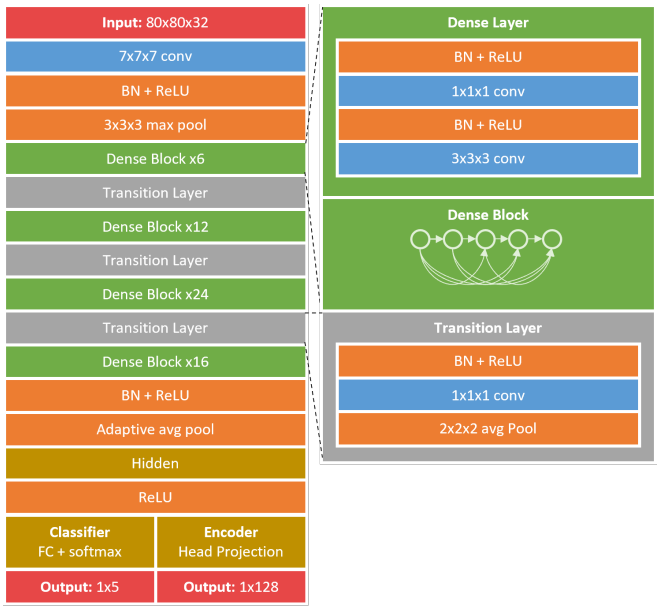


FIG. 4: Modified DenseNet121 architecture. It was used in two configuration modes: *encoder* (for pretraining) and *classifier* (for finetuning).

B. Using a pretrained network

The next experiment involved using the pretrained network on BHB-10K (see II A), then continue pretraining on NIMH and MultiRat_rest using an encoder configuration for SimCLR (see III A 1). After pretraining the model has been finetuned using cross entropy loss, similarly to the base model. Hyperparameters for finetuning were identical to the base model. For pretraining, the temperature parameter was set to 0.1, and the network has been trained for 50 epochs. The augmentation that was used for SimCLR was a random cutout of size $40 \times 40 \times 16$.

The epoch with the lowest value of NTXent loss on the validation set was number 45, and the generated model in that epoch has been used for the finetuning. The epoch that was chosen in the finetuning stage was 83, in a similiar fasion to the base model. The losses for the finetuning stage can be seen in Appendix C.

C. Adding contrastive learning on the dataset

Another interesting experiment to conduct is the comparison of SimCLR to SupCon to see how they aid to the model's performance.

Taking the pretrained model from the previous section (IV B), before the finetuning, another pretraining stage has been performed. This stage was done for the MaMI data itself, out of the notion that using contrastive learning will generate more distinct clusters in the embedding space. Only then the model has been finetuned to the same data.

Hyperparameters for both pretraining and finetuning were kept as they were in the previous sections, except for these experiments the pretraining stage was performed for a 100 epochs. The models were chosen based on the minimum in validation loss, as before.

After training both models, and looking at the results of the SimCLR model, another experiment has been made using a batch size of 32 instead of 10 for pretraining with SimCLR. This is because in contrastive learning, larger batch sizes provide more negative examples, facilitating convergence (i.e. taking fewer epochs and steps for a given accuracy). See FIG. 5

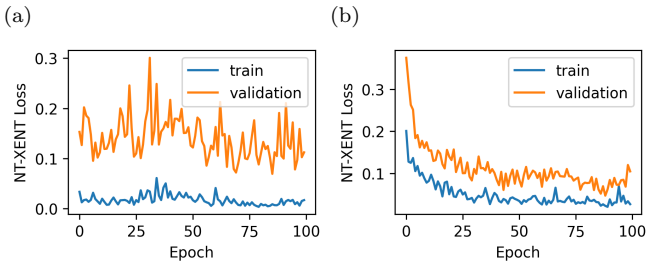


FIG. 5: NT-Xent loss for both SimCLR models. (a) - batch size 10. (b) - batch size 32.

D. Models comparison

Model accuracy scores can be seen at Table I. Latent space representation for each model are at FIG. 6 and

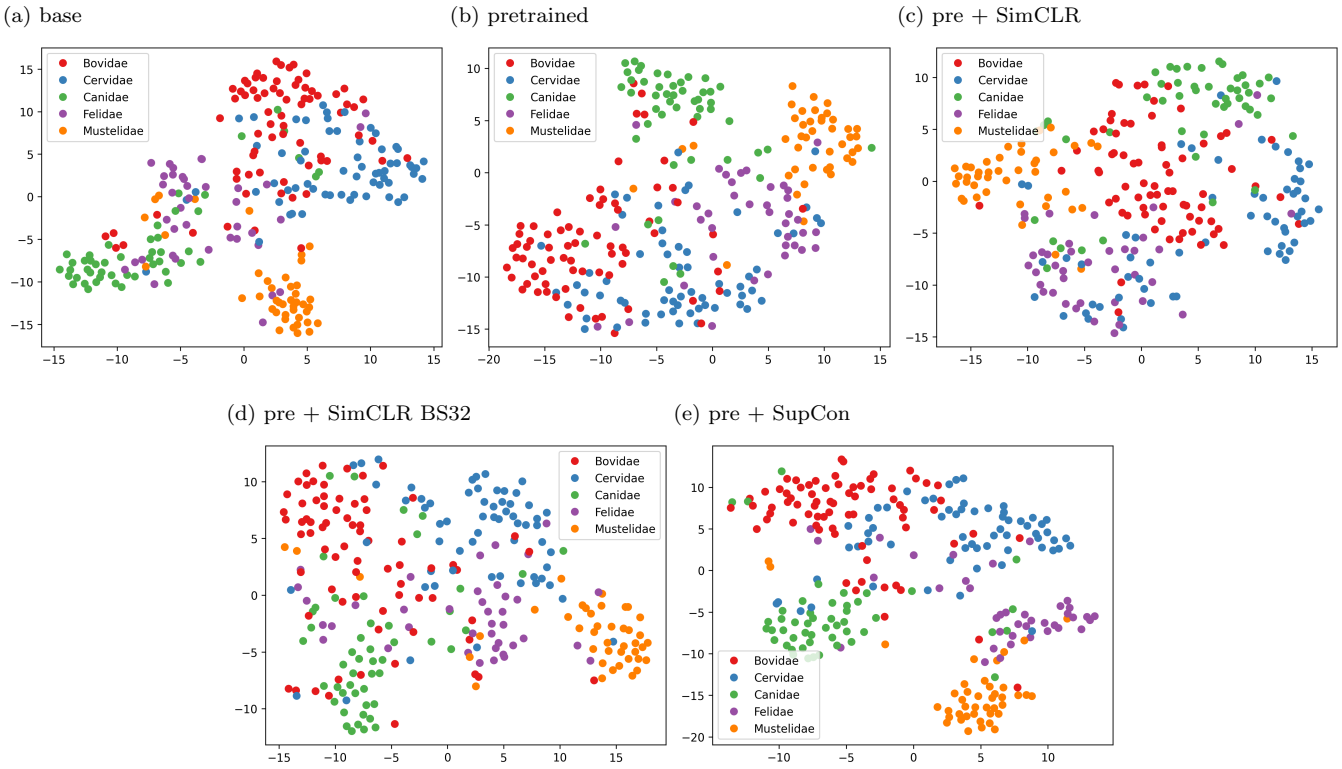


FIG. 6: Latent spaces representations for each of the models.

confusion matrices can be seen at Appendix D.

As expected, the model with the highest performance across almost all categories is the one trained using SupCon. It's APC and ACC scores were higher by around 8% than other models. However, the models trained with SimCLR were almost identical in their performance to the base model. That can be explained by the relatively small batch sizes comparing to the ones tested for this method (ones larger than 256). Furthermore, training for more epochs may have had an impact on the results, because more negative examples would have been introduced to the model, which had been shown to make an impact on the results [26].

E. Training on all classes

After selecting the best performing architecture, it has been decided to use it to train a model to classify all available mammalian families (a total of 12 families). This was done primarily to test how much data is required for a specific class to be represented well in the model.

Using the pretrained network, the network had been trained using SupCon loss with a batch size of 32 for a 100 epochs, and then finetuned using cross entropy loss for another 100 epochs. Minimum validation reached at epoch 98. After observing the results, the network was finetuned for another 100 epochs, to see if it will improve the results. Epoch 199 has been chosen. Comparison of the chosen model after 100 epochs and after 200 epochs can be seen in Table II. The latent space representation of the final model can be seen at FIG. 7.

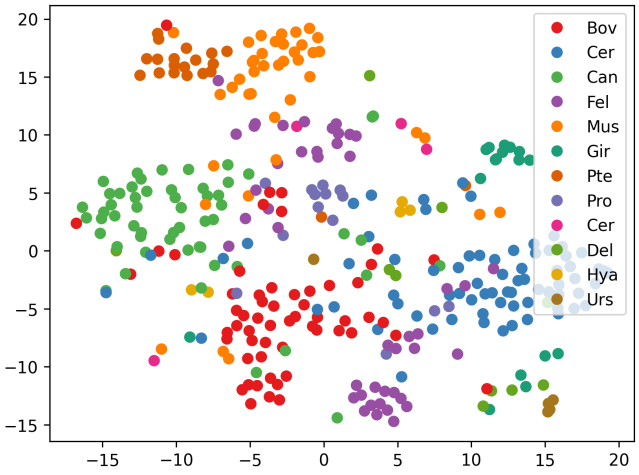


FIG. 7: Latent spaces representation of the model trained for all classes.

We can see a clear improvement in the results for the model trained for more epochs. A possible explanation for this phenomenon is that because there are more classes, more epochs need to be performed in order for

	Bov	Cer	Can	Fel	Mus	Gir	Pte	Pro	Cer	Del	Hya	Urs	Mean	Total
pre + SupCon BS32 98	**82.45**	59.72	62.5	**71.11**	**72.5**	82.35	45	42.85	0	**0**	0	100	51.54	63.95
pre + SupCon BS32 199	80.7	**75**	**78.57**	62.2	67.5	**94.11**	**75**	**50**	**50**	**0**	**80**	**100**	**67.76**	**72.09**

TABLE II: Accuracy scores for the models built for all classes, comparing the one built in 98 epochs and one in 199 epochs. Showing better performance for the model created after more epochs.

all permutations of the classes to appear on each batch. This will aid the contrastive elements of the algorithm to push different classes from each other, making the clusters more distinct in embedding space.

In the latent space visualization of the final model (FIG. 7) we can still observe the large clusters of the 5 most represented families, however they contain more noise in the form of the smaller clusters of the new families. Furthermore, class number 4 (Felidae) seems to be more spread out than before, which explains the relatively low ACC result compared to the previously trained models (almost 10% difference).

V. CONCLUSION

This project dealt with experimentation in the classification of mammalian brains using contrastive learning, on a fairly small and unbalanced dataset available. In order to first find the proper model training framework, it has been decided to experiment with the 5 most represented mammalian families. Given the small dataset and similarities between train and test samples, since some of them were generated from the same data point, even the base model's performance was quite decent (71.21% ACC on the test set).

Pretraining the network on larger datasets of human and rat cohorts seemed to have a positive effect on the results - an increase of 3% in ACC. However, adding SimCLR pretraining afterwards on the MaMI data negated this improvement. This can be explained by the relatively small batch sizes that were used for this algorithm due to memory constraints (10 and 32), comparing to sizes spanning from 256 to 8192 that were tested in the original paper. Due to the small batch sizes, perhaps it was needed to train the network for longer to supply it with enough negative samples that will naturally create clusters in the embedding space. Another important topic to consider in this area is the data augmentations. According to the original paper, no single transformation suffices to learn good representations [26], therefore adding more augmentations (like resized crops or rotations) to the training process might have assisted in clustering the data.

Using SupCon had a noticeable effect on the results - an improvement of 8% in overall ACC comparing to the base model. Aggregating both contrastive learning and the labels information assisted in clustering the data even on small batch sizes and less epochs, comparing to SimCLR. Furthermore, it performed better than the pretrained network which helps to indicate that it was indeed SupCon that had the effect. Looking at the latent space representation of the data (FIG. 6) we can see clear and more distant clusters for this model, presumably a result of the contrastive elements of the training.

After deciding on SupCon as the framework, another experiment for training the same model on all available families was run. Some of these samples only had one single image, so they suffered from under-representation during training. This was apparent in the results of the model, as some of these families had 0% ACC, while one class had a 100% accuracy, probably an overfitted result. Due to the higher number of classes, model convergence took longer than before, comparing checkpoints after a 100 or 200 epochs of training.

There are a few points to discuss regarding future work:

- Hyperparameter tuning could have aided in obtaining the highest performing model. This includes batch sizes, number of epochs to train (which I have discussed earlier), learning rate and weight decay (and perhaps using some scheduling method or a different optimizer altogether) and changing the temperature scalar used in the contrastive learning.

- It would be interesting to see if the network could learn the hierarchical representation of the taxonomy of these mammals. Meaning brains that belong to the same order will be closer in embedding space than ones that are from different orders. That can be achieved using some kind of multi-label classification algorithm, perhaps a novel weighted supervised contrastive loss [29]. This will also help with learning a clearer distance metric for mammalian brains.

- Another idea related to the distance metric may be to compare the results of an improved model trained on more data to a previous study on a connectomics-based taxonomy of mammals [30], seeing if the distance metric achieved by the model is similar to the one presented in the paper, which is that species that are part of the same taxonomic order tend to display similar connectome architecture.

- Using LIME or a similar feature visualization framework can be helpful in order to understand the features used by the network and maybe understand the most notable differences in brains of mammals.

Appendix A: Code links

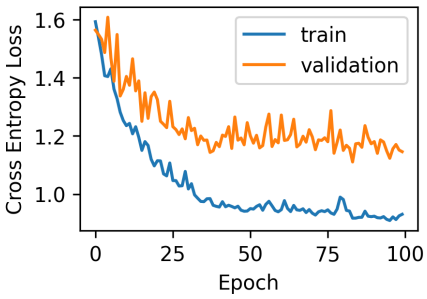
The code for this work is available here - `https://github.com/Idanlu/MaMIClass`.
Code was partially taken from:
`https://github.com/Duplums/yAwareContrastiveLearning`
`https://github.com/HobbitLong/SupContrast`.

Appendix B: MaMI dataset information

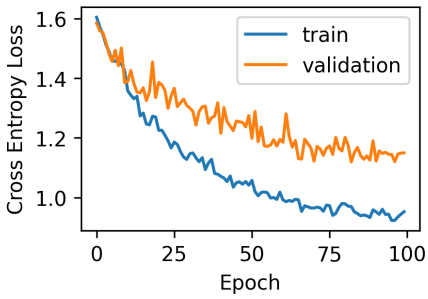
Name	Filename	Order	Superorder	Family
Addax	Addax	Artiodactyla	Laurasiatheria	Bovidae
Asian Otter	AsianOtter	Carnivora	Laurasiatheria	Mustelidae
Bactrian Deer	BactrianDeer1	Artiodactyla	Laurasiatheria	Cervidae
Bactrian Deer	BactrianDeer2	Artiodactyla	Laurasiatheria	Cervidae
Blanford Fox	BlanfordFox	Carnivora	Laurasiatheria	Canidae
Bottle Nose Dolphin	BottleNoseDolphin	Artiodactyla	Laurasiatheria	Delphinidae
Capra Nubiana	CapraNubiana1	Artiodactyla	Laurasiatheria	Bovidae
Capra Nubiana	CapraNubiana2	Artiodactyla	Laurasiatheria	Bovidae
Capra Nubiana	CapraNubiana3	Artiodactyla	Laurasiatheria	Bovidae
Capra Nubiana	CapraNubiana4	Artiodactyla	Laurasiatheria	Bovidae
Caracal	Caracal	Carnivora	Laurasiatheria	Felidae
Cat	Cat1	Carnivora	Laurasiatheria	Felidae
Cat	Cat3	Carnivora	Laurasiatheria	Felidae
Cat	Cat5	Carnivora	Laurasiatheria	Felidae
Cat	Cat6	Carnivora	Laurasiatheria	Felidae
Chital	Chital1	Artiodactyla	Laurasiatheria	Cervidae
Chital	Chital2	Artiodactyla	Laurasiatheria	Cervidae
Common Fox	CommonFox	Carnivora	Laurasiatheria	Canidae
Cow	Cow1	Artiodactyla	Laurasiatheria	Bovidae
Deer	Deer1	Artiodactyla	Laurasiatheria	Cervidae
Deer	Deer2	Artiodactyla	Laurasiatheria	Cervidae
Dik Dik	DikDik	Artiodactyla	Laurasiatheria	Bovidae
Dog	Dog1	Carnivora	Laurasiatheria	Canidae
Dog	Dog2	Carnivora	Laurasiatheria	Canidae
Dog	Dog3	Carnivora	Laurasiatheria	Canidae
Dog	Dog4	Carnivora	Laurasiatheria	Canidae
Dog	Dog5	Carnivora	Laurasiatheria	Canidae
Dog	Dog7	Carnivora	Laurasiatheria	Canidae
Fallow Deer	FallowDeer1	Artiodactyla	Laurasiatheria	Cervidae
Fallow Deer	FallowDeer2	Artiodactyla	Laurasiatheria	Cervidae
Fallow Deer	FallowDeer3	Artiodactyla	Laurasiatheria	Cervidae
Ferret	Ferret1	Carnivora	Laurasiatheria	Mustelidae
Ferret	Ferret2	Carnivora	Laurasiatheria	Mustelidae
Ferret	Ferret3	Carnivora	Laurasiatheria	Mustelidae
Ferret	Ferret4	Carnivora	Laurasiatheria	Mustelidae
Fruit Bat	FruitBat1	Chiroptera	Laurasiatheria	Pteropodidae
Fruit Bat	FruitBat2	Chiroptera	Laurasiatheria	Pteropodidae
Fruit Bat	FruitBat3	Chiroptera	Laurasiatheria	Pteropodidae
Giraffe	Giraffe1	Artiodactyla	Laurasiatheria	Giraffidae
Giraffe	Giraffe2	Artiodactyla	Laurasiatheria	Giraffidae
Giraffe	Giraffe3	Artiodactyla	Laurasiatheria	Giraffidae
Goat	Goat1	Artiodactyla	Laurasiatheria	Bovidae
Goat	Goat2	Artiodactyla	Laurasiatheria	Bovidae
Goat	Goat3	Artiodactyla	Laurasiatheria	Bovidae
Himalian Bear	HimalianBear	Carnivora	Laurasiatheria	Ursidae
Honey Badger	HoneyBadger	Carnivora	Laurasiatheria	Mustelidae
Hyaena	Hyaena2	Carnivora	Laurasiatheria	Hyaenidae
Hyrax	Hyrax1	Hyracoidea	Afrotheria	Procaviidae
Hyrax	Hyrax2	Hyracoidea	Afrotheria	Procaviidae
Leopard	Leopard2	Carnivora	Laurasiatheria	Felidae
Macaque	Macaque1	Primates	Euarchontoglires	Cercopithecidae
Cow	Cow	Artiodactyla	Laurasiatheria	Bovidae

Appendix C: Training and validation losses for finetuning stages

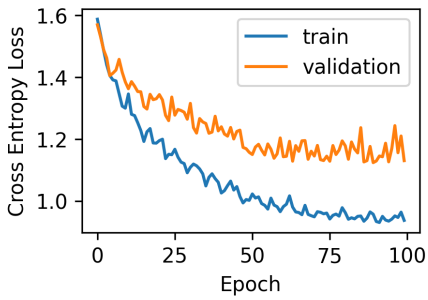
(a) base



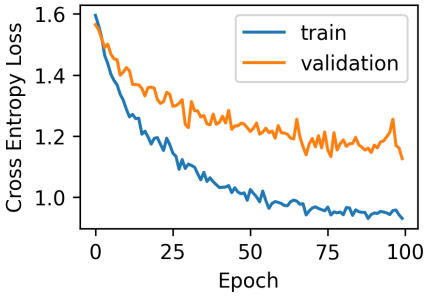
(b) pretrained



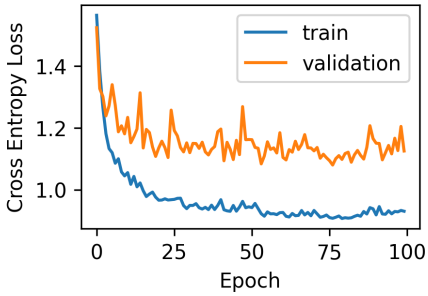
(c) pre + SimCLR



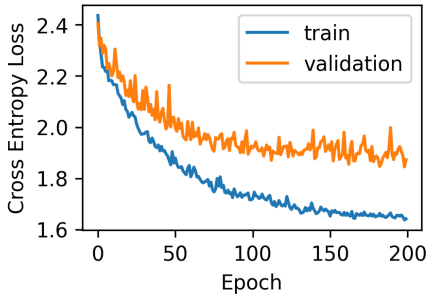
(d) pre + SimCLR BS32



(e) pre + SupCon



(f) pre + SupCon all classes



[1] S. Herculano-Houzel, Proceedings of the National Academy of Sciences **109**, 10661 (2012).

[2] C. S. von Bartheld, J. Bahney, and S. Herculano-Houzel, Journal of Comparative Neurology **524**, 3865 (2016).

[3] P. Rea, *Essential Clinical Anatomy of the Nervous System* (Elsevier, 2015).

[4] P. Hagmann, EPFL 10.5075/EPFL-THESIS-3230 (2005).

[5] J. M. Segall, E. A. Allen, R. E. Jung, E. B. Erhardt, S. K. Arja, K. Kiehl, and V. D. Calhoun, Frontiers in Neuroinformatics **6**, 10.3389/fninf.2012.00010 (2012).

[6] G. Katti, S. Ara, and D. Shireen, Intl J Dental Clin **3** (2011).

[7] A. Murphy and F. Gaillard, Mri sequences (overview) (2015).

[8] S. Gassenmaier, T. Küstner, D. Nickel, J. Herrmann, R. Hoffmann, H. Almansour, S. Afat, K. Nikolaou, and A. E. Othman, Diagnostics **11**, 2181 (2021).

[9] A. S. Lundervold and A. Lundervold, Zeitschrift für Medizinische Physik **29**, 102 (2019).

[10] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, Medical Image Analysis **35**, 18 (2017).

[11] L. Zhang, M. Wang, M. Liu, and D. Zhang, Frontiers in Neuroscience **14**, 10.3389/fnins.2020.00779 (2020).

[12] Y. Gao, Z. Li, C. Song, L. Li, M. Li, J. Schmall, H. Liu, J. Yuan, Z. Wang, T. Zeng, L. Hu, Q. Chen, and Y. Zhang, Physics in Medicine & Biology **66**, 10.1088/1361-6560/abd2c5 (2021).

[13] Y. Assaf, A. Bouznach, O. Zomet, A. Marom, and Y. Yovel, Nature Neuroscience **23**, 805 (2020).

[14] P. S. Ulinski, The cerebral cortex of reptiles, in *Comparative Structure and Evolution of Cerebral Cortex, Part I*, edited by E. G. Jones and A. Peters (Springer US, Boston, MA, 1990) pp. 139–215.

[15] L. E. Suarez, Y. Yoval, M. P. van den Heuvel, O. Sporns, G. Lajoie, and B. Misic, Mami dataset (2022).

[16] B. Dufumier, P. Gori, J. Victor, A. Grigis, M. Wessa, P. Brambilla, P. Favre, M. Polosan, C. McDonald, C. M. Piguet, M. Phillips, L. Eyler, and E. D. and, in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021* (Springer International Publishing, 2021) pp. 58–68.

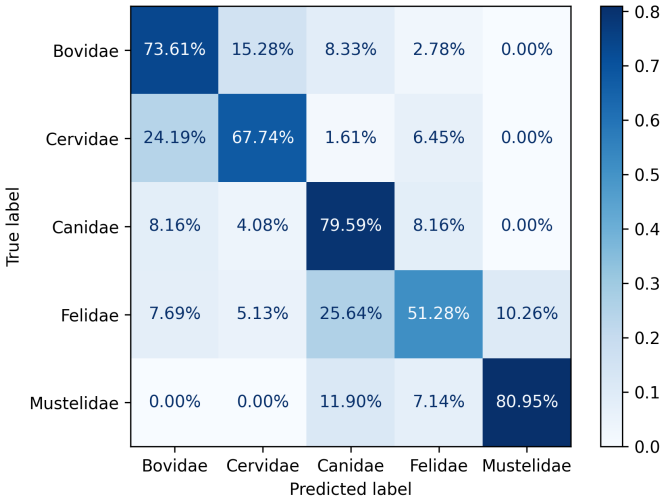
[17] A. C. Nugent, A. G. Thomas, M. Mahoney, A. Gibbons, J. T. Smith, A. J. Charles, J. S. Shaw, J. D. Stout, A. M. Namyst, A. Basavaraj, E. Earl, T. Riddle, J. Snow, S. Japee, A. J. Pavletic, S. Sinclair, V. Roopchansingh, P. A. Bandettini, and J. Chung, Scientific Data **9**, 10.1038/s41597-022-01623-9 (2022).

[18] A. C. Nugent, A. G. Thomas, M. Mahoney, A. Gibbons, J. Smith, A. Charles, J. S. Shaw, J. D. Stout, A. M. Namyst, A. Basavaraj, E. Earl, T. Riddle, J. Snow, S. Japee, A. Pavletic, S. Sinclair, V. Roopchansingh, P. A. Bandettini, and J. Chung, 10.18112/openneuro.ds004215.v1.0.1 (2022).

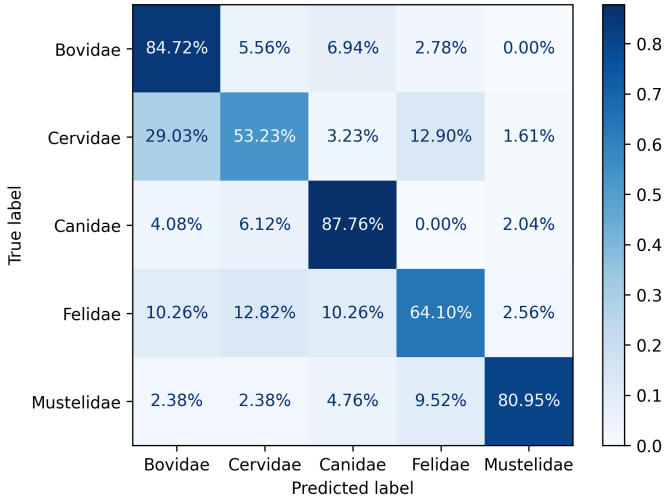
[19] J. Grandjean, G. Desrosiers-Gregoire, C. Anckaerts, D. Angeles-Valdez, F. Ayad, D. A. Barrière, I. Blockx, A. B. Bortel, M. Broadwater, B. M. Cardoso,

Appendix D: Confusion matrices for selected models for 5 classes

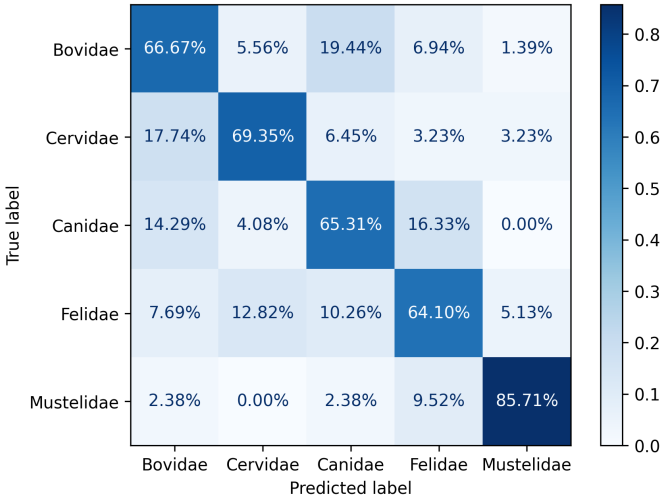
(a) base



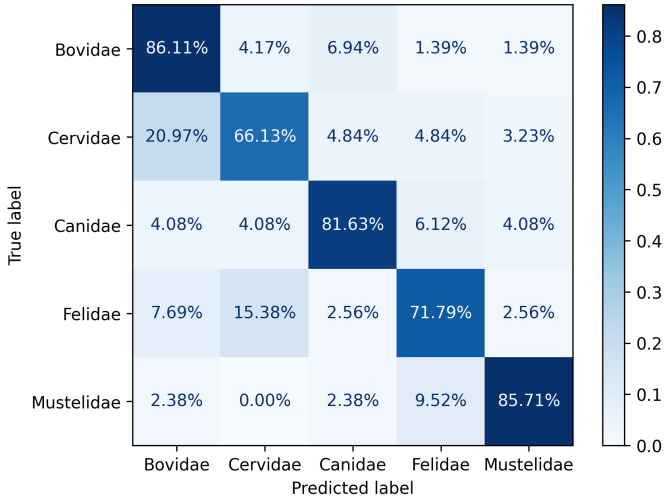
(b) pretrained



(c) pre + SimCLR BS32



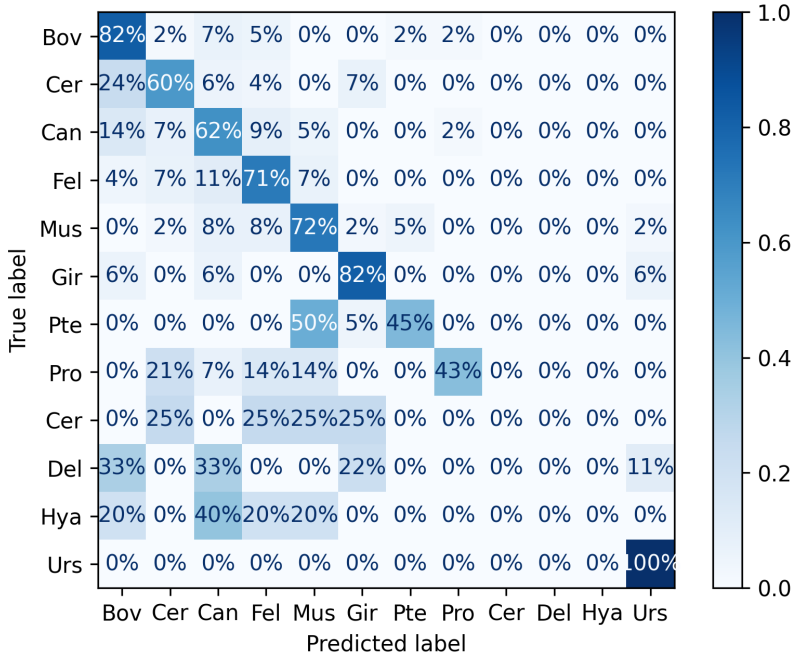
(d) pre + SupCon



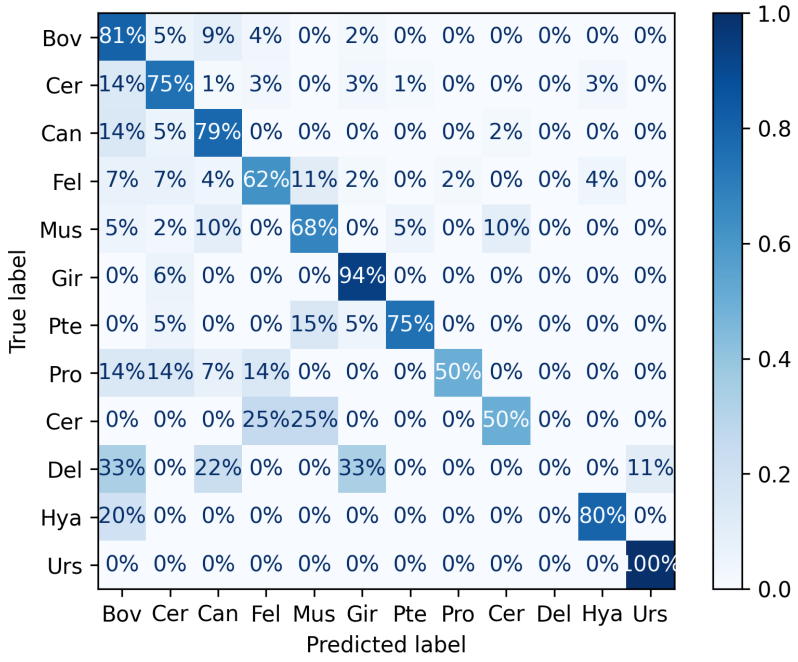
M. Célestine, J. E. Chavez-Negrete, S. Choi, E. Christiaen, P. Clavijo, L. Colon-Perez, S. Cramer, T. Daniele, E. Dempsey, Y. Diao, A. Doelemeyer, D. Dopfel, L. Dvoáková, C. Falfán-Melgoza, F. F. Fernandes, C. F. Fowler, A. Fuentes-Ibañez, C. Garin, E. Gelderman, C. E. Golden, C. C. Guo, M. J. Henckens, L. A. Hennessy, P. Herman, N. Hofwijks, C. Horien, T. M. Ionescu, J. Jones, J. Kaesser, E. Kim, H. Lambers, A. Lazari, S.-H. Lee, A. Lillywhite, Y. Liu, Y. Y. Liu, A. L. Castro, X. López-Gil, Z. Ma, E. MacNicol, D. Madularu, F. Mandino, S. Marciano, M. J. McAuslan, P. McCunn, A. McIntosh, X. Meng, L. Meyer-Baese, S. Missault, F. Moro, D. Naessens, L. J. Nava-Gomez, H. Nonaka, J. J. Ortiz, J. Paasonen, L. M. Peeters, M. Pereira, P. D. Perez, M. Pompilus, M. Prior, R. Rakhmatullin, H. M. Reimann, J. Reinwald, R. T. de Rio, A. Rivera-Olvera, D. Ruiz-Pérez, G. Russo, T. J. Rutten, R. Ryoke, M. Sack, P. Salvan, B. G. Sanganahalli, A. Schroeter, B. J. Seewoo, E. Selingue, A. Seuwen, B. Shi, N. Sirmpilatze, J. A. Smith, C. Smith, F. Sobczak, P. J. Stenroos, M. Straathof, S. Strobelt, A. Sumiyoshi, K. Takahashi, M. E. Torres-García, R. Tudela, M. van den Berg, K. van der Marel, A. T. van Hout, R. Vertullo, B. Vidal, R. M. Vrooman, V. X. Wang, I. Wank, D. J. Watson, T. Yin, Y. Zhang, S. Zurbruegg, S. Achard, S. Alcauter, D. P. Auer, E. L. Barbier, J. Baudewig, C. F. Beckmann, N. Beckmann, G. J. Becq, E. L. Blezer, R. Bolbos, S. Boretius, S. Bouvard, E. Budinger, J. D. Buxbaum, D. Cash, V. Chapman, K.-H. Chuang, L. Ciobanu, B. Coolen, J. W. Dalley, M. Dhenain, R. M. Dijkhuizen, O. Esteban, C. Faber, M. Febo, K. W. Feindel, G. Forloni, J. Fouquet, E. A. Garza-Villarreal, N. Gass, J. C. Glennon, A. Gozzi, O. Gröhn, A. Harkin, A. Heerschap, X. Helluy, K. Herfert, A. Heuser, J. R. Homberg, D. J. Houwing, F. Hyder, G. D. Ielacqua, I. O. Jelescu, H. Johansen-Berg, G. Kaneko, R. Kawashima,

Appendix E: Confusion matrices for the models for all classes

(a) pre + SupCon BS32 98



(b) pre + SupCon BS32 199



S. D. Keilholz, G. A. Keliris, C. Kelly, C. Kerskens, J. Y. Khokhar, P. C. Kind, J.-B. Langlois, J. P. Lerch, M. A. López-Hidalgo, D. Manahan-Vaughan, F. Marchand, R. B. Mars, G. Marsella, E. Micotti, E. Muñoz-Moreno, J. Near, T. Niendorf, W. M. Otte, P. Pais, W.-J. Pan, R. A. Prado-Alcalá, G. L. Quirarte, J. Rodger, T. Rosenow, C. S. Baptista, A. Sartorius, S. J. Sawiak, T. W. Scheenen, N. Shemesh, Y.-Y. I. Shih, A. Shmuel, G. Soria, R. Stoop, G. J. Thompson, S. M. Till, N. Todd, A. V. D. Linden, A. van der Toorn, G. A. van Tilborg, C. Vanhove, A. Veltien, M. Verhoye, L. Wachsmuth, W. Weber-Fahr, P. Wenk, X. Yu,

V. Zerbi, N. Zhang, B. B. Zhang, L. Zimmer, G. A. Devenyi, M. M. Chakravarty, and A. Hess, 10.18112/openneuro.ds004114.v1.0.0 (2022).
[20] M. Jenkinson, C. F. Beckmann, T. E. Behrens, M. W. Woolrich, and S. M. Smith, NeuroImage **62**, 782 (2012).
[21] M. Jenkinson, M. Pechaud, S. Smith, *et al.*, in *Eleventh annual meeting of the organization for human brain mapping*, Vol. 17 (Toronto., 2005) p. 167.
[22] Y. Liu, H. S. Unsal, Y. Tao, and N. Zhang, Neuroinformatics **18**, 395 (2020).
[23] P. McCarthy, Fsleyes (2022).
[24] N. Saunshi, J. Ash, S. Goel, D. Misra, C. Zhang, S. Arora, S. Kakade, and A. Krishnamurthy, arXiv e-prints 10.48550/ARXIV.2202.14037 (2022).
[25] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, Advances in Neural Information Processing Systems **33**, 18661 (2020).
[26] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, arXiv e-prints , arXiv (2020).
[27] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, Densely connected convolutional networks (2016).
[28] Z. Zhong, M. Zheng, H. Mai, J. Zhao, and X. Liu, Journal of Physics: Conference Series **1651**, 012143 (2020).
[29] L. Zheng, J. Xiong, Y. Zhu, and J. He, in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2022) pp. 2594–2604.
[30] L. E. Suárez, Y. Yovel, M. P. van den Heuvel, O. Sporns, Y. Assaf, G. Lajoie, and B. Misic, bioRxiv 10.1101/2022.03.11.483995 (2022).