Proposition of a New Experiment to Better Understand the Relation Between Typicality and Prototypes

Samuel Kostadinov
University of Trento
Trento, Italy
samuel.kostadinov@studenti.unitn.it

***Abstract*—The typicality is a topic that still needs exploration. In this paper, the proposed experiment makes use of feature extraction to build a prototype and a siamese network to find a possible correlation between the typicality of a concepts and the similarity between a given image and the built prototype.**

***Index Terms*—Typicality, Prototypes, CNNs, Siamese Network**

I. Introduction

The typicality of a concept is a topic that needs a lot of exploration, since it's difficult to evaluate precisely the typicality of an object and also because the people's brains are always a little different from each other. The experiment I would like to propose, has the objective of making us understand better the bond between the perceived typicality of an object that belongs to a category and the prototype of that category. This can help the scientists to better understand how knowledge is organized in the brain. This experiment consists in different phases, such as data collection, feature extraction, prototype construction and similarity judgment.

II. Background

A. Feature extraction

Feature extraction is a topic widely studied. This topic has numerous different applications. One of its goals is to reduce the amount of computational power needed for image processing. There are various techniques to extract meaningful features from images. Some of them are very common and very easy to implement. These are, for example, edge extraction or shape analysis. Other possibilities, more advanced, involve neural networks.

1) Classic feature extraction techniques: One of the first feature extraction methods implemented in image processing is edge detection. In 2019, a paper by Owotogbe presented a review of edge detection techniques. These are usually divided into two groups, gradient-based and Gaussian-based. Some examples are the Sobel operator and the Canny edge detector. Each of the methods has pros and cons. It's the user's job to find the most appropriate for its goal [1]. After edge detection, other techniques began to spread for feature extraction. In a paper by Kumar and Kumar Bhatia written in 2014 [2], there are some examples of techniques used to extract features. In the first one, the authors present different types of features and then some techniques to extract them. These are:

- **Diagonal based feature extraction techniques**:
  In this procedure, the image is divided into zones formed by small squares of pixels. In the paper's case, there would be 19 diagonal lines. The value of each pixel in these diagonals is summed to obtain a single sub feature. Then we can extract a feature by averaging the sub features. With this method, we can extract a feature for every zone. Then by averaging the column-wise and row-wise features we can increase their number.
- **Fourier descriptors**:
  The Fourier transform is commonly used for shape analysis. The Fourier transformed coefficient form the Fourier descriptors. These descriptors represent the shape in a frequency domain, with the low frequencies symbolizing the general shape and the high frequencies symbolizing details of the shape. Since the transformation usually generates many parameters, only a subset is considered.
- **Principal Component Analysis (PCA)**:
  This procedure is a mathematical way to convert a set of observations into a set of values of uncorrelated variables. These variables are defined so that the first one has the highest variance, and the components are all orthogonal (independent) from each other.
- **Independent component analysis (ICA)**:
  ICA is a statistical technique. It aims to use non-Gaussian random variables to represent multidimensional vectors. The random variables should be as independent as possible.
- **Gabor filter**:
  A Gabor is a sinusoid multiplied by a Gaussian, and its response is a convolution operation. This type of filter performs well in both spatial and frequency domains.
- **Chain Code Histogram of Character Contour**:
  This method is based on a contour following technique.

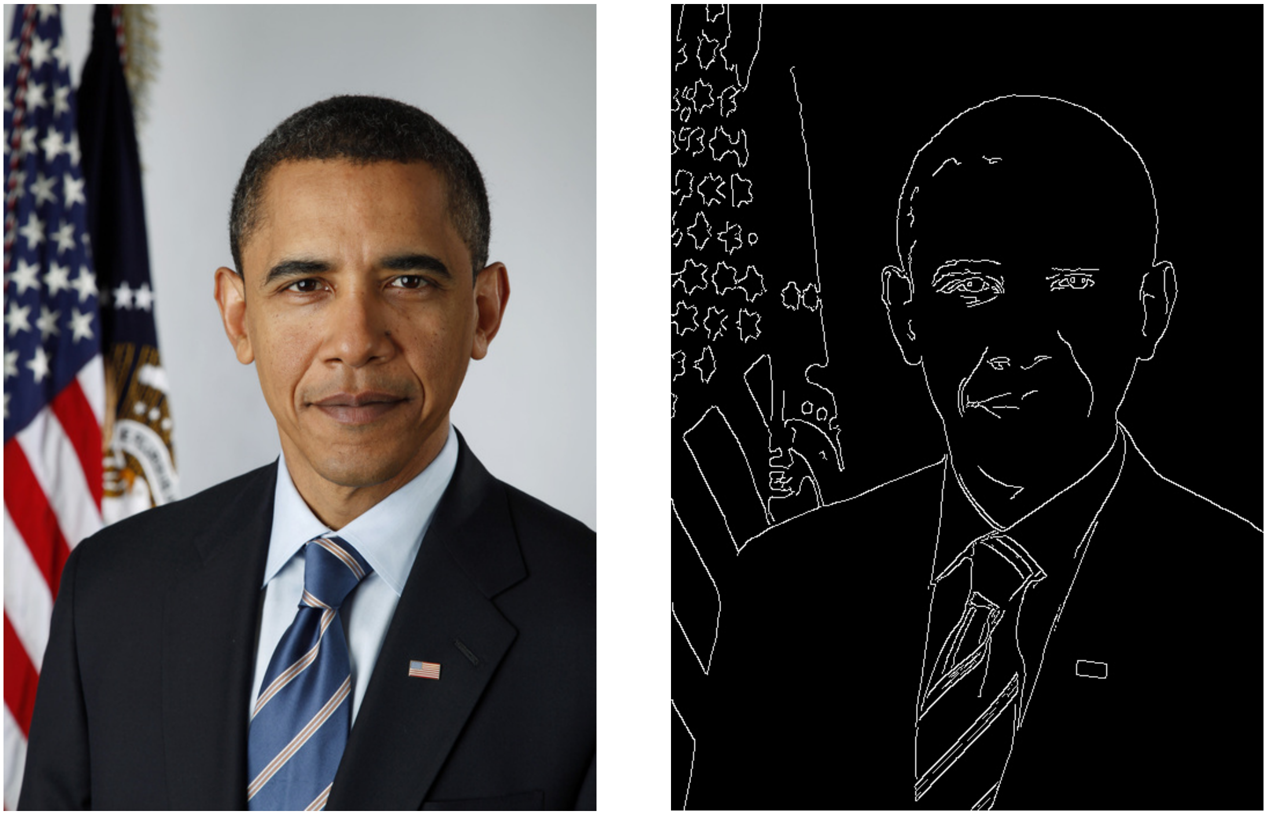

Fig. 1: Example of edge detection

The contour following uses a chain coding standard proposed by Freeman, that assigns a value to every pixel to identify the next pixel in the border.

- **Finding Intersection/Junction in character**:
  Using the same standard proposed by Freeman used in Chain Code Histogram, there is the possibility to count intersections (or junctions) and open ends in a figure.
- **Transition feature**:
  This method is based on the transition from background to foreground. There are different techniques that can work both on gray level images or in 4-connected or 8-connected images.

These were not the only techniques presented in the paper, but since some were more problem-specific (about handwritten character recognition) were excluded from the list. These excluded techniques can still be used, but may have worse results or may require some adaptation. The techniques were: Fractal theory techniques, Shadow features of character, Sector approach for feature extraction, Extraction of distance and angle features, Extraction of occupancy and end-point features, and Zernike moments.
In another paper, written in 2013 by Tian [3], there are some other techniques cited that may result useful:

- **colour features**:
  colour is one of the most important features humans can perceive. The features we can extract depend on the colour space, but once it's defined there are some possibilities. A few examples are colour histograms, colour moments and colour coherence vectors. One of the simple and meaningful, according to the authors, is colour moments. The most common colour moments are the mean, the standard deviation and the skewness.
- **Texture features**:
  Another very important feature of images is texture. Features involving texture analysis can be extracted from groups of pixels. One of the most common methods is a Gabor filter, which can be used by characterizing the central frequency and the orientation parameters.

2) Deformable shape analysis: A more sophisticated technique than the ones listed before is shape analysis. Deformable shape analysis, in particular, can be useful to extract features from an image.
In 2005, Felzenszwalb wrote a paper that focuses on representation and detection of deformable shapes [4]. For the goal of the experiment proposed here, some deformable shape detection techniques can be useful.
The technique proposed strongly relies on the triangulated polygon representation. This type of representation lets approximate every 2D shape without holes using a representation based on triangles. The author also make use of the properties of chordal graphs and k-trees.
The technique described in this paper falls under the category of deformable template matching. One of the components of the method is the energy function, a function that associates

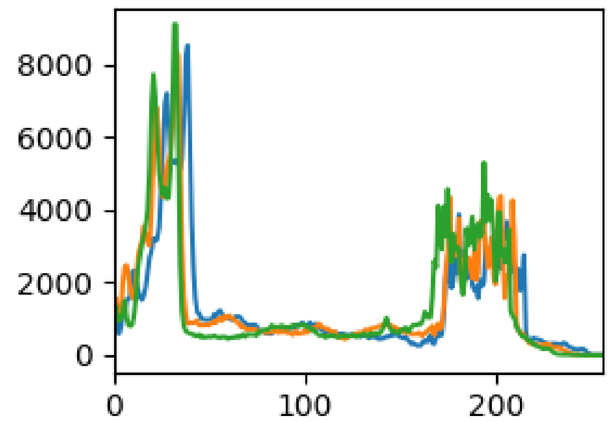

Fig. 2: Example of color histogram

a cost with every possible transformation. The objective is to find the transformation with the lowest possible cost. This component is very flexible, since depending on the formulation of the energy function the costs can be tuned even for individual triangles. Moreover, it is also possible to integrate learning techniques to learn deformation parameters. Since the possible non-rigid transformations of a template are numerous, this kind of techniques usually requires an initialization near to the correct solution, although this is not required for the algorithm presented in this paper. For the implementation of the algorithm itself, a techniques called non-serial dynamic programming was the key factor. In fact, using the order of perfect elimination (property of the every triangulated simple polygon), the algorithm computes the optimal position of every vertex with respect to the other vertices. Once the algorithm solves for the last vertex, it can update all the others as in typical dynamic programming and obtain the optimal location for every vertex.

The learning of the parameters can be seen as looking for the location with the highest. The matching problem is then fed to a statistical framework that computes the configuration minimizing the energy cost, which corresponds to the best template match.

3) Machine learning based feature extraction techniques: After the classic techniques, machine learning began to be involved in feature extraction. In particular there are some papers that explain the results achieved with machine learning.

In 2019, Halimi et al. wrote a paper about a feature extraction technique that they implemented using unsupervised deep learning [5]. Although this paper examines the techniques for 3D shapes, it's very likely that it can be adapted to extract 2D features from images.

The background of this experiments is formed of some concepts that are not immediate. One of the concepts used as background is the minimum distortion correspondence. The other are, for example, descriptor learning, functional maps

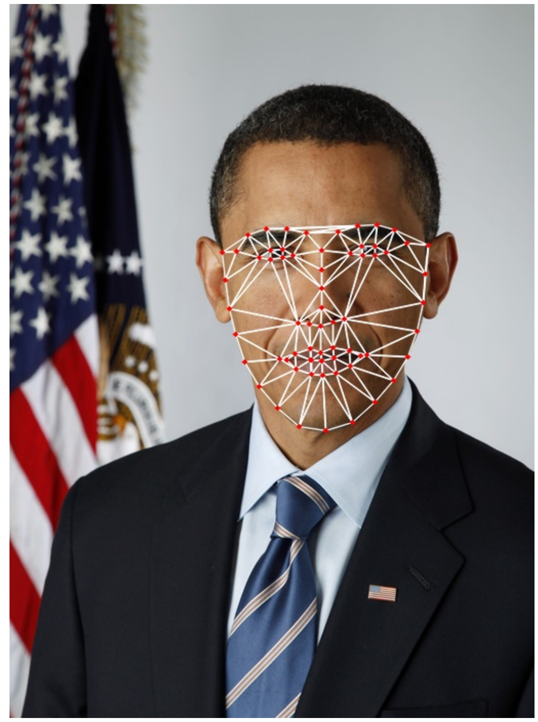

Fig. 3: Example of delaunay triangulation, the type of triangulation at the base of Felzenszwalb's paper

and deep functional maps. These concepts serve as a base for all the work done later.

In particular, the authors focus on unsupervised deep functional maps. The authors state that the main contribution of their paper is that they introduce a technique with which they can avoid annotating a lot of data before training a model. After reducing the number of vertices to a number between 5000 and 7000, the authors used a deep functional map network. The inputs are fed to two residual neural networks, which compute a dense descriptor fields. These descriptors, are the inputs to the functional map layer. Lastly, as post processing, a point-wise map recovery was applied and, after that, the shape was upscaled to the original resolution.

In 2019, there is another paper, written by Varshni et al. that proposes another method [6]. This paper has the objective to classify images suing CNNs to understand if a patient has pneumonia or not. After the preprocessing, in which the authors just resized the images, there is the feature extraction process. For this step, the authors used different pretrained models. In this paper's results it's reported that the best model architecture is DenseNet-169. This kind of neural network, called Dense Nets, overcomes the problem of gradient vanishing. The model used in particular for the work of this paper had 4 dense blocks and 3 transition layers. Then there is a classification layer.

In 2019, Guamei et al. wrote a paper about hybrid feature extraction methods to classify brain tumours [7]. The process described in the paper is divided into three points. The first one is preprocessing, the second is feature extraction and the

third is brain tumour classification. The first step is just to rescale the values to the range $[0,\ 1]$. The feature extractor used is the GIST descriptor. The GIST is then combined with a Gabor filter and produces a total of $mxnx4x4$ GIST feature vectors. There is a variant of the GIST descriptor, the PCA-NGIST is a PCA-based normalized GIST feature extraction method. The NGIST is a normalized version of the GIST descriptor, that uses the L2 norm to make the GIST invariant to illumination and shadowing. After this step, the only thing left is the classification of the tumour, which was performed with a RELM classifier.
In Figure 4 there is an example of feature extracted by a Convolutional Neural Network.

4) Hyperspectral images feature extraction: Although there exists techniques that can extract features from hyperspectral images, this kind of images are created to capture the entire spectrum. This fact means that these images can capture more information then what the human eye can. techniques involve, in the majority of the cases, learning of some form, especially in form of neural networks, in particular convolutional recurrent networks, as seen in a paper by Hu et al. [8] and in a paper by Rasti et al. [9]. For the reasons just said, it's probably more meaningful to use normal images than these ones.

B. Networks

Neural Networks are a powerful instrument in the hands of computer scientists. There exists a lot of different types of artificial neural networks, each with its own peculiarities. In particular, for this experiment the one needed will be, most likely, just convolutional neural networks and siamese neural networks.

1) Convolutional Neural Network (CNN): Convolutional neural networks are not an extremely recent type of neural network, especially if we consider that LeNet-5 was created in 1998. Scientists have dealt with CNNs for some years. Now, as of 2022, CNNs are mostly used to deal with images.
In 2017, Wu wrote a paper that had the purpose to serve as an introduction to CNNs [10]. These networks use mostly three kind of layers: Dense layers (which are fully connected layers), convolution layers and pooling layers. In the following, there will be a short description of each kind of layer.

- **Dense layer**:
  The dense layer is not peculiar of CNNs, in fact it's found in almost all types of neural networks. It consists of a set of neurons that are connected with every neuron of the previous layer. Dense layers are usually used towards the end of the network.
- **Convolutional layer**:
  Convolutional layers just compute a convolution operation. This helps reducing the input size and extract meaningful features from the image taken as input. These layers are usually used in the first layers of the neural network.
- **Pooling layers**:
  Pooling layers downsample the image to reducing the number of parameters needed. These layers are usually alternated with the convolution layers, so are used at the beginning of a neural network.

The applications of deep learning and CNNs are numerous. As shown previously, there are some CNNs used in image classification with the objective to identify diseases. The typical example is the pneumonia detection. Another possibility is to use CNNs in segmentation or self driving cars, as stated in [11] by Alzubaidi et al.
In figure 5 there is an example of the architecture of a CNN, in particular AlexNet is the CNN chosen for this example.

2) Siamese Neural Network: Siamese neural networks, also called twin networks, are a more recent advance in computer science. These neural networks have a different structure from convolutional neural networks. The structure of siamese neural networks is peculiar. In fact, there are two identical networks, called embedding networks, that process different inputs and, after that, merge in a single layer that measures the similarity between the two outputs of the two networks. The siamese neural networks have a different application compared to CNNs, in fact twin networks are mainly used to measure similarity of two objects. In the case of this paper, the siamese neural networks are used for image matching. This is not the first time that siamese networks are used for image matching, as proved by a paper written in 2016 by Melekhov et al. [12], but with a little difference in the objective of the matching. The objective of the paper was to learn a general similarity for image retrieval, while on the case of this paper, the objective is to find a similarity measure with a prototype.
In Figure 6 there is an example of a Siamese Neural Network, in particular the two embeddings are convolutional neural networks.

C. Similarity judgments

In the past there have been some works regarding image similarity, like for example the one presented by Appalaraju and Chaoji in 2017 [13]. In this work they explore the possibility to find a similarity measure for image retrieval. To achieve this goal, they implemented a siamese neural network that uses two identical multi-scale CNNs that share weights. The ouput of the network is binary, being 1 the label to predict in case of negative examples and 0 the label to predict in case of positive example. The loss function used in this work is a contrastive loss. This paper then proceeds to explain a technique called Online Pair Mining Strategy (OPMS), inspired by curriculum learning, that helps reducing the training time.
This is not the only work that uses neural networks to learn similarity, in fact there are also other application fields for similarity study. One of them is healthcare, as shown in a paper by Suo et al., written in 2017, that introduces a model used to make personalized prediction of diseases for patients, basing the prediction on a similarity related method, learned using CNNs [14]. In this paper, before learning the actual similarity measure, the authors processed the input through a CNN and performed a step they call “Time fusion”. This step let them incorporate the time information and weight differently

```
[[17.47104     0.          0.         ... 14.563998  14.728311   3.9745955]
 [ 0.          5.2031355   0.         ...  1.8009549   0.          9.976197 ]
 [ 0.7887765   6.159361    2.5253575 ...  2.1835113 13.718938   3.5711129]
 ...
 [ 1.3785725   0.         15.821116  ...  0.          7.099158    8.3597    ]
 [13.986625    0.         15.957546  ...  0.          0.          0.        ]
 [ 0.          0.         12.463112  ...  0.          2.4566994   0.8278876]]
```

Fig. 4: Example of features extracted by a CNN

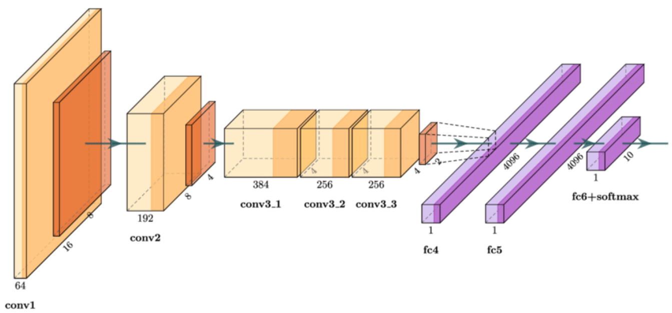

Fig. 5: Example of CNN architecture. In particular, this image represents AlexNet's architecture

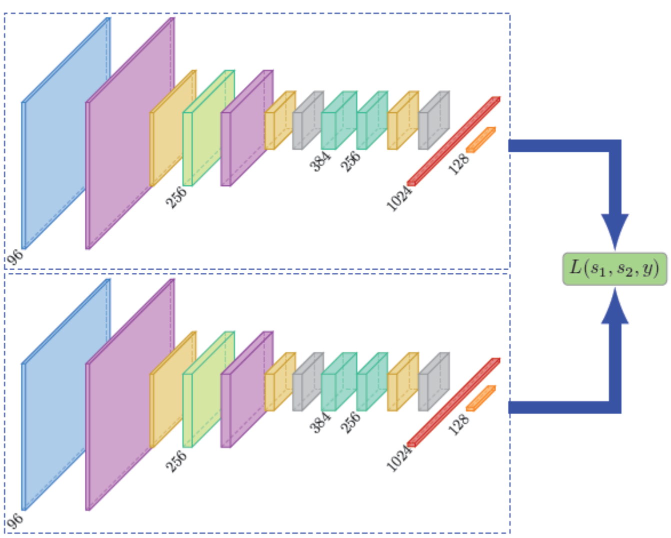

Fig. 6: Example of a Siamese Network architecture.

eventual symptoms based on their temporal distance from the onset of the disease. Only after this step the authors proceeded to learn a similarity metric, basing the learning on CNNs.
Another paper, written by Chopra et al. in 2005, used a method similar to the one written by Appalaraju. In this work, the authors used a siamese architecture to learn a similarity metric and applied it in the field of face verification [15].
More interestingly, there are also other papers, in particular a paper by Attarian et al. written in 2020 uses $VGG16$. This paper's goal is to use this CNN to predict human similarity judgments [16]. Using $VGG16$, the authors were able to achieve good results, with over 90% in validation accuracy.

III. Data Collection

A. Image collection

Since the proposed experiment uses neural networks, there is the need to have a lot of data. In this particular case, there is the need to gather many images. There are some image dataset, intended for computer vision [1], that could help. Some of the dataset reported there are quite famous, like ImageNet or Google's Open Images. The best for this kind of work, though, is to have a dataset that bases every image on the same "macro-category", like for example a dataset made of images of birds.

B. Typicality values

In this experiment, typicality judgments play a crucial role. To gather enough data about the perceived typicality of images, a form is necessary. This survey should be structured in a way that presents an image ad a time to the subject that is taking the survey and asks to rate the typicality of the object portrayed in the image, possibly in a given range. If that's the case, a range between 1 and 10 could be the best decision since people are quite used to rate in this range. Once there are enough results, the various rates should be averaged so that there is a single measure that takes into account every rate taken from the survey. The number of people that should participate in the experiment should be high enough to be a meaningful statistical group.

IV. Feature Extraction

While collecting the needed data with the survey, it's possible to begin the feature extraction procedure. There are various possibilities for this step, as shown in the Section dedicated to feature extraction background. It could be interesting to test both classical and machine learning methods, to measure which one is the most efficient. Since feature extraction has already been described in the paper, this section will present a discussion about the results of the presented methods' results. The main difference between the classical and the machine learning feature extraction techniques is human interpretability. In fact, for classical feature extraction techniques, humans have no difficulty in interpreting the result of the process. For example, in edge extraction, humans have no difficulty

[1] https://imerit.net/blog/22-free-image-datasets-for-computer-vision-all-pbm/

in recognizing if the edge extracted is correct or not seeing the original image. For example, in Figure 1 no one has any problem to recognize that the figure on the right is the one that draws the edges of the figure on the left, while in Figure 4 there is no clear correlation between the feature and what it means in the image itself. Moreover, classical and machine learning features can give different weights to different features. In fact, classical methods rely mostly on features that humans consider important, on the basis of general experience about the world, but also based on the way our brain works. An empirical confirmation of this fact can be that if a person asks to another person to describe, for example, a common kingfisher, the person that should describe the bird will, most likely, start describing the dimension, the shape, and only later the colour. Machine learning techniques, on the other hand, could give a very high importance to the fact that the bird is mainly blue and red, if they have access to the colour information. Due to the fact that the blue colour is not very present in nature, the blue and red couple of colours can be highly characteristic of the common kingfisher, even if humans give, unconsciously and innately, more importance to shape-related features.

V. Prototype construction

The construction of the prototype is one of the most delicate parts of the experiments. How this phase works depends heavily on what method was used for feature extraction.

A. Classical feature extraction techniques

In case classical feature extraction techniques were used, the goal of this activity is to build an image that presents in itself a "prototype" of the object. For example, a prototype of a bird, could be an imaginary bird with a shape that it's obtained with a weighted average of the shapes of all the birds in the image. For example, since a birds like sparrows and swallows are perceived as more typical than ostriches and penguins the prototype should resemble more swallows and sparrows than ostriches and penguins. To do so, the key factor is performing numerous analyses on various different subjects, until there are enough data to have a statistical value.
In Figure 7 there is an example of the edges of two different birds. Although both of the subjects of the original subjects are birds, one is much more typical than the other one. In this case, the prototype should try to match more the swallow edge than the penguin edge. A way to do so could make use deformable template matching. Matching a shape with a template could be useful to find a weighted average between the two shapes. In this way, the resulting shape takes into account both the original edges while giving priority to the one more typical. This procedure should be applied for both shape features and colour features, so that all the features that can be extracted from the prototype image resemble the weighted average of all the feature from the original images.

B. Machine learning feature extraction techniques

In case machine learning feature extraction techniques were used, it's conceptually and practically easier to compute the average features. In fact, the features extracted by machine learning techniques are in the form of arrays of numbers, so to compute the prototype of the image in this form it's enough to take the weighted average of the vectors.

C. Considerations about prototype construction

The prototype depends on the feature extraction technique that was used. As discussed in Section IV, the key difference is human interpretability. In fact, building a prototype using the classical feature extraction methods, results in an image that can be seen and interpreted. Using a prototype generated with machine learning feature extraction techniques results in an array of numerical values, that are not easily interpretable. As of 2022, there has been some research on the topic of interpretability and explainability of machine learning, like for example a work by Carvalho written in 2019 [17], but the field is still growing, also given the interest deriving from the AI-Act, the proposition of the european union in the field of artificial intelligence. In particular, there is a document of the act [2] that states that there will be some requirements in terms of transparency and intepretability for machine learning systems that are considered high-risk AI. The growing interest produced some results, but still they're not enough to easily interpret machine learning features.
Most likely, in the future will be possible to look at the features extracted by a neural network and comprehend their meaning. When this will be possible, independently of the feature extraction techniques used, the human interpretability will be possible, and with that in mind it would be probably more convenient to extract the features using neural networks. The reason behind this consideration is that although the training of a neural network can be expensive in computational terms, the evaluation of an input on a trained network it's not very computationally demanding. On the other side, algorithms like the triangulation always have the same demand, that is lower than the demand of training a neural network, but surely higher than the demand of evaluating a single input on a neural network. Moreover, considering the possiblity to use transfer learning techniques, the training can become a fine-tuning or a domain adaptation, which means the computational load is lighter. Considering that the fact that the training is a demanding task to be perfomed only once and that other algorithms may be less demanding but must be run multiple times, it's probably more convenient to train a neural network. Moreover, not only the feature extraction process is convenient, but also the prototype construction itself is more convenient if the features are in the form of numerical parameters, since it's enough to perform a weighted average, a simpler operation if confronted with the construction of a new image that should take into account features of numerous other images.

[2] https://eur-lex.europa.eu/resource.html?uri=cellar:e0649735-a372-11eb-9585-01aa75ed71a1.0001.02/DOC_1&format=PDF

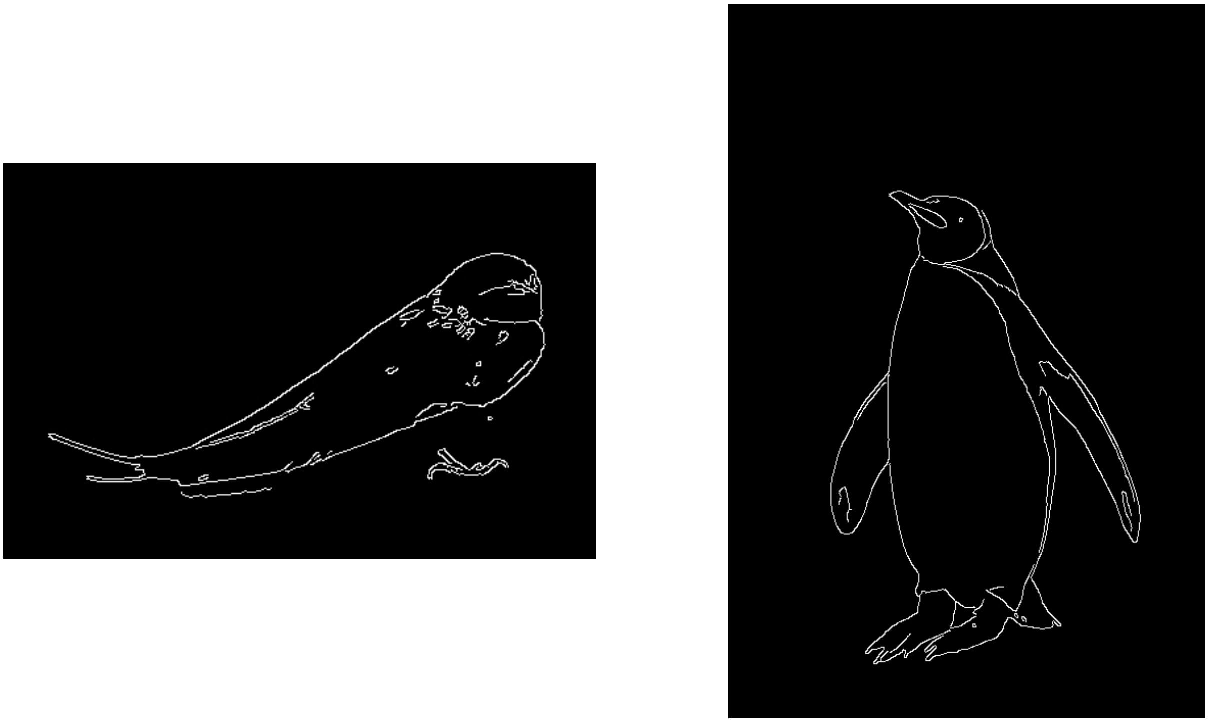

Fig. 7: Edge extraction of swallow and penguin

VI. Similarity learning

The similarity learning is the core part of the experiment. In this experiment, the similarity learning is done through a siamese neural network but, as the previous steps, depends on the feature extraction techniques used.

A. Classic feature extraction techniques

If the feature extraction process happened with classical techniques, after the prototype building step the result is an image. This means that the two embedding networks must be able to process images. Most likely the best choice, in this case, is to implement two CNNs as embedding networks and merge them in a single fully connected network for the siamese similarity learning. In this case, the siamese network would be fed the prototype image and an image from the dataset. The goal of this kind of prediction is to learn a similarity metric with some annotated values. After learning the similarity metric, the network will be able to measure the similarity of the prototype image and any other image fed in the other embedding network. By doing so and retrieving various measurements the couple of values *perceived typicality - similarity measure* could reveal some interesting patterns.

B. Machine learning feature extraction techniques

If the feature extraction process involved neural networks, after the prototype building step the result is not anymore an image but it's a vector of values. This implies that there is no need to use a convolutional neural network as embedding, but it's enough to have some fully connected layers that will be merged in a single fully connected network for the similarity learning. The rest of the procedure is similar, since the siamese network will still be fed the features of the prototype and the features of an imaage of the dataset. As in the previous case, the goal of this kind of prediction is to learn a similarity metric using annotated values. After learning the metric, the network will be able to measure the similarity of the prototype and any other image. By doing so the couple of values *perceived typicality - similarity measure* could reveal some interesting patterns.

C. Loss function

The loss function is a key factor of this kind of leaning, togheter with the distance measure. In fact, siamese neural networks usually use a loss function that is called contrastive loss. This kind of loss is based on a distance measure that should be defined by who is conductiing the experiment. The type of measure that it's possible to use ranges a lot, from simple distances like the manhattan distance or the euclidean distance to more complex measures that can be defined by the user.

D. Considerations about similarity learning

The two possible embedding networks, convolutional and fully connected, lead to different internal results but the final goal is the same. In fact, even if with different computations, the siamese network outputs a similarity measure. This output strongly relies on the distance measure. In fact, the same two vectors can have different similarity measures, based on the type of distance that the user decided to implement. For example, suppose that the vectors are $[2,\ 3]$ and $[5, 8]$. If the manhattan distance is chosen as measure, the distance between the two defined vectors is 8. Using the euclidean distance instead of the manhattan distance, the result is $5,83$. In data science there are some possibilities for distance measurements, a part from euclidean and manhattan distance. Some examples are the cosine distance or the Hamming distance. Moreover, there could be an ad hoc distance measure, defined for this problem in order to optimize the results obtained by the neural network. This kind of freedom is very beneficial, since it gives the possibility to use different formulas and, seeing the results, there is also the possibility to see which distance works better. This could give an insight on which one of the distance measures is the closest to the one used by human brain in similarity judgement.

Focusing more on the computational aspect of the similarity learning, the more convenient approach is the one involving neural networks feature extraction. In fact, if the features are extracted with neural networks, the siamese network will only need fully connected layers, while it would need convolutional, pooling and fully connected layers if the features are extracted with in a classical way. The only problem about the machine learning feature extraction techniques is, as mentioned in V-C, is the fact that we're still unable to interpret the features extracted by a CNN.

VII. Possible Further Experiments

A. Possible variants in similarity learning

The siamese network described in VI-D has no constraints about both the distance measure and the loss function. An interesting possibility could be using a distance measure that is sensitive to human similarity judgments. In this way, the network will optimize the weights in order to predict based on the human similarity measure. The resulting similarity measure can be, in this case, compared with the state-of-art human similarity judgments obtained with CNNs.

Another possibility is to adopt a variant of the previously described network. For example, it's possible to use a variant of siamese neural networks that uses triplets for training. In this version, the network takes in input a reference (which is the prototype in the case of this experiment) and two values, one positive and one negative. The network tries to maximize the distance between the reference and the negative example, while minimizing the distance between the reference and the positive example. There is a way, as shown by Dor et al. in [18], to adapt the triplet network to learn a similarity metric. In this way, the similarity metric is learned and so the network itself tries to optimize it. There are different research papers that show how triplet networks usually outperform siamese networks, especially in one-shot learning. Could be interesting, in this experiment, to confront both siamese neural networks and triplet networks to see which performs better. Moreover, this could give some hints on how the human similarity judgments work.

B. Possible relevant experiments

One of the first ideas that is an interesting follow-up of this experiment is to compare the results obtained with the two possible feature extraction methods. Even though one of the two methods is not interpretable by humans, it's still interesting trying to understand which of the methods is more efficient in detecting features relevant for the similarity learning. Moreover, if the results indicate that the machine learning techniques are more efficient, could be a push in the research that try to understand the neural networks' feature extraction method. This could be interesting for the future of neural networks and neuroscience.

After the described experiment, there are some natural extensions of the experiment. One of the most natural and obvious extensions of the experiment is to try the same experiment with another category of images. For example, if the first experiment is conducted on a dataset with bird images, it could be interesting to conduct the same experiment on a dataset that uses a different animal, like, for example, fishes.

Another experiment that comes as natural is trying to apply some transfer learning techniques. This could be interesting because, as far as we know, the human brain similarity can work in different ways in judging similarity of different images. Applying transfer learning, we could see if the feature extracted for a category are also effective for a different category. This can give extra insight about the typicality and its perception and how it works in human brain. The results can show that human brain innately use the same kind of measure of typicality or a different measure of typicality for different classes. Of course, either result should be investigated more, since a single test is not enough to prove that one of the two possibilities is more likely than the other. Even though it's not enough for a proof, could be an interesting starting point for neuroscientists to better understand how typicality works in human brain.

Another possibly interesting topic could be examining the results of the survey based on the region of the person answering. In fact, it's possible that different regions, that have slightly different cultures and that may result in slightly different typicality perception. For example, in Europe the eagles are not common, so it's very likely that people will give a high typicality to eagles, while in North America, since the bald eagle is a national symbol, it's easier that people associate with it a higher typicality value. Moreover, there are a lot of different factors that can influence the typicality rates of images in people. For example, a farmer could be more inclined than people that live in a big city to think associate a chicken with the concept of "bird", and thus giving it a higher typicality.

Lastly, it's worth notice that in this paper the prototype is considered as the mean of the features extracted from all the the images of the dataset. A possibility that could be interesting would be interesting to try to experiment the possibility to use different concept of prototype and implementing the correspondent methods of building it.

VIII. Conclusion

The experiment proposed has the objective to help identifying, if it exists, a link between the typicality of a concept and the similarity to a prototype. The work strongly relies on neural networks, in particular siamese networks, and using them tries to achieve the goal of linking similarity of a prototype with the typicality. Both if the siamese network can achieve good statistics or if it cannot achieve good statistics, the result can give an idea of the correlation between typicality and a prototype of a concept.

References

[1] JS Owotogbe, TS Ibiyemi, and BA Adu. "Edge detection techniques on digital images-a review". In: *Int J Innov Sci Res Technol* 4 (2019), pp. 329–332.

[2] Gaurav Kumar and Pradeep Kumar Bhatia. "A detailed review of feature extraction in image processing systems". In: *2014 Fourth international conference on advanced computing & communication technologies*. IEEE. 2014, pp. 5–12.

[3] Dong ping Tian et al. "A review on image feature extraction and representation techniques". In: *International Journal of Multimedia and Ubiquitous Engineering* 8.4 (2013), pp. 385–396.

[4] Pedro F Felzenszwalb. "Representation and detection of deformable shapes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.2 (2005), pp. 208–220.

[5] Oshri Halimi et al. "Unsupervised learning of dense shape correspondence". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4370–4379.

[6] Dimpy Varshni et al. "Pneumonia detection using CNN based feature extraction". In: *2019 IEEE international conference on electrical, computer and communication technologies (ICECCT)*. IEEE. 2019, pp. 1–7.

[7] Abdu Gumaei et al. "A hybrid feature extraction method with regularized extreme learning machine for brain tumor classification". In: *IEEE Access* 7 (2019), pp. 36266–36273.

[8] Wen-Shuai Hu et al. "Spatial–spectral feature extraction via deep ConvLSTM neural networks for hyperspectral image classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 58.6 (2020), pp. 4237–4250.

[9] Behnood Rasti et al. "Feature extraction for hyperspectral imagery: The evolution from shallow to deep: Overview and toolbox". In: *IEEE Geoscience and Remote Sensing Magazine* 8.4 (2020), pp. 60–88.

[10] Jianxin Wu. "Introduction to convolutional neural networks". In: *National Key Lab for Novel Software Technology. Nanjing University. China* 5.23 (2017), p. 495.

[11] Laith Alzubaidi et al. "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions". In: *Journal of big Data* 8.1 (2021), pp. 1–74.

[12] Iaroslav Melekhov, Juho Kannala, and Esa Rahtu. "Siamese network features for image matching". In: *2016 23rd international conference on pattern recognition (ICPR)*. IEEE. 2016, pp. 378–383.

[13] Srikar Appalaraju and Vineet Chaoji. "Image similarity using deep CNN and curriculum learning". In: *arXiv preprint arXiv:1709.08761* (2017).

[14] Qiuling Suo et al. "Personalized disease prediction using a CNN-based similarity learning method". In: *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2017, pp. 811–816. DOI: 10.1109/BIBM.2017.8217759.

[15] Sumit Chopra, Raia Hadsell, and Yann LeCun. "Learning a similarity metric discriminatively, with application to face verification". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. IEEE. 2005, pp. 539–546.

[16] Maria Attarian, Brett D Roads, and Michael C Mozer. "Transforming neural network visual representations to predict human judgments of similarity". In: *arXiv preprint arXiv:2010.06512* (2020).

[17] Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. "Machine Learning Interpretability: A Survey on Methods and Metrics". In: *Electronics* 8.8 (2019). ISSN: 2079-9292. DOI: 10.3390/electronics8080832. URL: https://www.mdpi.com/2079-9292/8/8/832.

[18] Liat Ein Dor et al. "Learning Thematic Similarity Metric from Article Sections Using Triplet Networks". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 49–54. DOI: 10.18653/v1/P18-2009. URL: https://aclanthology.org/P18-2009.

[19] Stefan Schneider, Graham W. Taylor, and Stefan C. Kremer. "Similarity Learning Networks for Animal Individual Re-Identification - Beyond the Capabilities of a Human Observer". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*. Mar. 2020.