

# Classification Of Skin Lesions By Topological Data Analysis Alongside With Neural Network

Naierreh Elyasi <sup>1</sup> and mehdi hosseini moghadam <sup>2</sup>

<sup>1</sup>Affiliation not available

<sup>2</sup>Kharazmi University

October 30, 2023

## Abstract

In this paper we use TDA mapper alongside with deep convolutional neural networks in the classification of 7 major skin diseases. First we apply kepler mapper with neural network as one of its filter steps to classify the dataset HAM10000. Mapper visualizes the classification result by a simplicial complex, where neural network can not do this alone, but as a filter step neural network helps to classify data better. Furthermore we apply TDA mapper and persistent homology to understand the weights of layers of mobilenet network in different training epochs of HAM10000. Also we use persistent diagrams to visualize the results of analysis of layers of mobilenet network.

,

# Classification Of Skin Lesions By Topological Data Analysis Alongside With Neural Network

N. Elyasi <sup>a,\*,1</sup>, M. Hosseini Moghadam <sup>b,2</sup>

<sup>a</sup>Department of math and computer Sciences, Kharazmi University, Taleghani street, Tehran, Iran

<sup>b</sup>Department of Engineering, Kharazmi University, Taleghani street, Tehran, Iran

## ARTICLE INFO

### Keywords:

Mapper

Neural network

Persistent Homology

Topological Data analysis

## ABSTRACT

In this paper we use TDA mapper alongside with deep convolutional neural networks in the classification of 7 major skin diseases. First we apply kepler mapper with neural network as one of its filter steps to classify the dataset HAM10000. Mapper visualizes the classification result by a simplicial complex, where neural network can not do this alone, but as a filter step neural network helps to classify data better. Furthermore we apply TDA mapper and persistent homology to understand the weights of layers of mobilenet network in different training epochs of HAM10000. Also we use persistent diagrams to visualize the results of analysis of layers of mobilenet network.

## 1. Introduction

The incidence of skin cancer and skin problems in world has increased dramatically over the last few years. Despite preventative public health measures, rates are continuing to hit new records. Skin cancer can be divided to some major types like melanoma, Basal cell carcinoma (BCC) and Squamous cell carcinoma (SCC).

However skin cancer grows very quickly but the process of treatment would be much easier and also faster if it can be detected fast enough. With proper and accurate machine learning models we can detect skin cancer with high accuracy.

In the last few years many papers has been published aiming on the detection of the skin cancer with remarkable results, for more information we encourage readers to check out [13].

Artificial neural networks specially deep neural networks are among the best algorithms for image classification hence skin lesions, however there is an issue with deep neural networks, namely the black box problem. Here we try to explain the black box problem with an example: Imagine a doctor after several observations and experiments came to the decision that a patient has skin cancer, this decision has been made in an special frame work based on some predefined rules, for instance at the first step the patient must take test1, if the result of test1 is positive then he must take test 2 and so on; at the end after taking several such tests doctor came to his decision. But when a neural net classifies a patient as a person who has cancer there is no such bench mark or frame work in order to understand why it came to this decision despite the fact that its decision was exact and accurate. In deep learning we call this problem the black box problem [2]. In this paper we present some ways to solve this problem.

Data science is the field of science in which it is struggling with huge and messy amount of data in order to turn

it to something useful. One of the main difficulties in data science is to deal with high dimensional data and transform it into data with less dimensionality in order to make it easier for analyzing.

Topological data analysis (TDA) is one of the brand-new and fast growing fields of data science which is trying to analyze data by studying its shape and also reducing the dimensionality of data [3]. TDA is based on two very important branches of mathematics Statistics and Algebraic Topology, because of its methodology TDA can solve some serious problems in data science. The goal of TDA is to reduce the dimensionality of high dimensional data and also analyzing the topological structure or shape of data and finally clustering complex data. TDA also provides innovative data mining methods that can improve the efficiency of machine learning techniques. Two of best algorithms in TDA are persistent homology and Mapper. In Persistent homology, a filtration of combinatorial objects, simplicial complexes, is constructed and then main topological structures of data is derived. Some visualization tools such as "Persistent Diagram", "Barcode" and "Persistent Landscape" are invented to indicate the main topological features of data. Persistent homology has been previously used in brain [6], image analysis [4] and data mining [5].

Goal of TDA-Mapper is to convert a high-dimensional data into a combinatorial object called simplicial complex. Simplicial complex tries to sum up the topological structure of the data as it reduces the dimensionality of data. TDA Mapper has been previously applied to classify clinical [8], [10], [11].

Automated classification of skin lesions using images is a challenging task because of the structure of skin images.

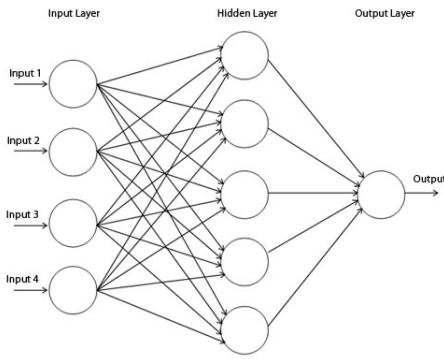
In this paper first we demonstrate classification of skin lesions using TDA Mapper from images directly, using only pixels and disease labels as inputs. We use neural networks as one of filters in mapper to gain better results. Second we will explain how topological data analysis can address the problem of being black box, especially obtaining insight into how convolutional neural networks (CNNs) work.



elyasi82@khu.ac.ir (N.E.); m.h.moghadam1996@gmail.com (M.H.

Moghadam)

ORCID(s):



**Figure 1:** single hidden layer neural network

Also we introduce some backgrounds about artificial intelligence and novel methods in TDA. Then we describe the kaggle data set HAM10000 that consists images of eight different types of skin diseases by sketching some diagrams like chord diagram and heatmap to compare the relation between different type of skin cancer and gender or position of cancer. Next we apply mapper alongside a neural network to classify our dataset(HAM10000). At last we analyse the weights of the layers of mobilenet neural network that is trained with HAM10000 dataset and visualize the results of this analysis by the means of mapper and persistent diagrams.

## 2. Preliminaries

### AI and neural networks

Artificial intelligence (AI) is a branch of computer science inspired by human, human brain and its ability to learn new concepts and solve problems. Artificial neural networks (ANN) are the implementation of biological neurons, ANN learn to perform tasks and solve problems by considering and experiencing examples called "DATA", ANN generally do this with out any programming tasks or any other predefined rules.

In our brain neurons has organized in layers and information and biological signals transfer from one layer to another. Based on this architecture ANNs also made up of several different layers. inspired by biological neural networks, ANN contains of the following 3 types of layers:

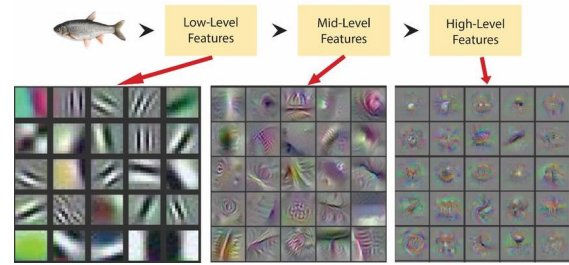
- Input layer which is used to feed input data to neural net(an image, text or any suitable type of data).
- Hidden layers that are the layers in between the input and output layers, hidden layers do the most computational tasks related to an artificial neural net. These layers are responsible for learning, also the mapping between input and output.
- Output layer which give us the result of the model (classification or regression).

Typically, a neural network is initially trained or fed with large amounts of data. Training consists of providing input

and telling the network what the output should be with respect to the given input.

By giving outputs to the network, it will update its weights in order to get the right predictions.

A related problem is that there is often a certain kind of overfitting to particular data sets. For such reasons, it is important to develop methods for developing some understanding of the internal states of the neural networks, for example in figure 2 we can see the patterns extracted in different layers of a neural net. Because of the very large number of nodes (or neurons) in the networks, this becomes a problem in data analysis, specifically for unsupervised data analysis.



**Figure 2:** Patterns extraction in different layers of a neural net

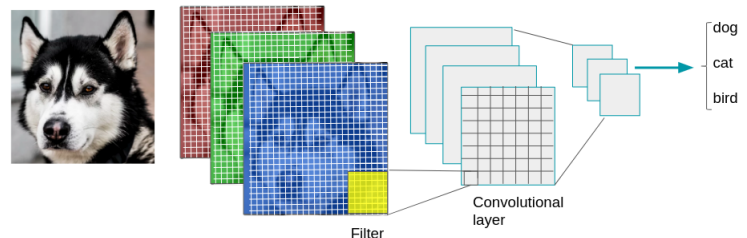
### 2.1. Convolutional Neural Networks

Convolutional Neural Networks (CNN) are class of neural networks which has been designed to work with visual data like image.

As discussed earlier simple neural nets receive an input and takes it through a series of hidden layers. Each hidden layer is made up of a set of neurons, where each neuron is fully connected to all neurons in the previous layer.

Simple neural nets don't scale with images with high number of pixels, because if we consider a simple 3-channel (RGB) image with  $32 \times 32$  pixel size then the input vector would be in the size of  $3 \times 32 \times 32 = 3072$ ; however  $32$  by  $32$  pixel image is almost out of date and there is no such image today.

In CNN the layers are organised in 3 dimensions: width, height and depth in addition to that, the neurons in one layer do not connect to all the neurons in the next layer but only to a small fraction of them.



**Figure 3:** Simple convolutional neural network for classification

In the case of a CNN, the convolution operation is performed on the input data (pixels of an image) with the use of a filter to then produce a feature map. We execute a convolution by sliding the filter over the input. At every location, a matrix multiplication is performed and sums the result onto the feature map. Figure 19 below shows the mechanism of convolution operation. You can see the filter is sliding over our input and the sum of the convolution goes into the feature map. The area of our filter is also called the receptive field the size of this filter is 3x3.

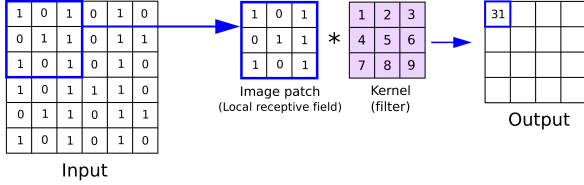


Figure 4: Single CNN with filter size of 3

## 2.2. Pooling

Another tool that CNNs use is called pooling, figure 5. Pooling is a way to take large images and shrink them down while preserving the most important information in them. It consists of stepping a small window across an image and taking the maximum value from the window at each step. In practice, a window 2 or 3 pixels on a side and steps of 2 pixels work well. After pooling, an image has about a quarter as many pixels as it started with. Because it keeps the maximum value from each window, it preserves the best fits of each feature within the window. This means that it does not care so much exactly where the feature fit as long as it fit somewhere within the window.

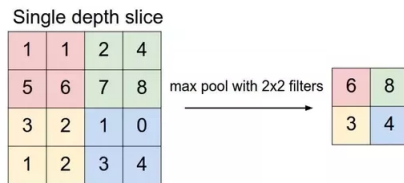


Figure 5: A pooling layer with 2 by 2 filter size

## 2.3. Depthwise Convolution

In depthwise convolution, we use each filter channel only at one input channel. In figure 6 we have 3 channel filter and 3 channel image. The algorithm first breaks the filter and image into three different channels and then convolve the corresponding image with corresponding channel and then stack them back, next it uses the 1 by 1 convolutional filters called pointwise convolution in the context of depthwise separable convolution. The benefit of depthwise convolution is that it has less computations than regular convolution.

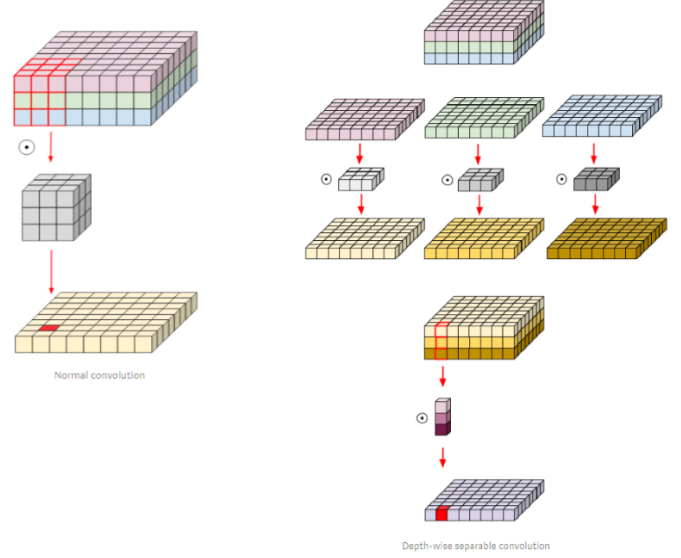


Figure 6: In this depthwise convolution we have 3 channel filter and 3 channel image. The algorithm first breaks the filter and image into three different channels and then convolve the corresponding image with corresponding channel and then stack them back, next it uses the 1 by 1 convolutional filters called pointwise convolution in the context of depthwise separable convolution

## 2.4. Inception network vs Mobilenet

Mobilenets [7] are class of efficient models for mobile and embedded vision applications. MobileNets are based on a streamlined architecture that uses depth-wise separable convolutions to build light weight deep neural networks to deal with resource and accuracy tradeoffs and it has shown strong performance compared to other popular models on ImageNet classification such as Inception. Since Mobilenet has depthwise convolution instead of regular convolution it is lighter and faster to train and it can be used for mobile applications. In figure 7 and 8 you can see the architecture of both Mobilenet and Inception.

## 2.5. Clustering

Clustering is the task of mapping set of objects into classes called as clusters, there is more similarity in objects of one cluster than objects in two different clusters. Clustering is an unsupervised classification of data points into classes or clusters. Steps in clustering algorithm are as follows:

- First of all dimensionality and features of the given data will be checked by methods such as feature selection or feature extraction.
- Next checking the similarity of data points like Euclidean Distance, Mean Square etc.
- Next the data points are grouped together into clusters, based on similarity measures obtained from the previous step.
- Finally data is represented with compact description of individual cluster and further more the cluster pro-

Table 1. MobileNet Body Architecture

| Type / Stride | Filter Shape                         | Input Size                         |
|---------------|--------------------------------------|------------------------------------|
| Conv / s2     | $3 \times 3 \times 3 \times 32$      | $224 \times 224 \times 3$          |
| Conv dw / s1  | $3 \times 3 \times 32 \text{ dw}$    | $112 \times 112 \times 32$         |
| Conv / s1     | $1 \times 1 \times 32 \times 64$     | $112 \times 112 \times 32$         |
| Conv dw / s2  | $3 \times 3 \times 64 \text{ dw}$    | $112 \times 112 \times 64$         |
| Conv / s1     | $1 \times 1 \times 64 \times 128$    | $56 \times 56 \times 64$           |
| Conv dw / s1  | $3 \times 3 \times 128 \text{ dw}$   | $56 \times 56 \times 128$          |
| Conv / s1     | $1 \times 1 \times 128 \times 128$   | $56 \times 56 \times 128$          |
| Conv dw / s2  | $3 \times 3 \times 128 \text{ dw}$   | $56 \times 56 \times 128$          |
| Conv / s1     | $1 \times 1 \times 128 \times 256$   | $28 \times 28 \times 128$          |
| Conv dw / s1  | $3 \times 3 \times 256 \text{ dw}$   | $28 \times 28 \times 256$          |
| Conv / s1     | $1 \times 1 \times 256 \times 256$   | $28 \times 28 \times 256$          |
| Conv dw / s2  | $3 \times 3 \times 256 \text{ dw}$   | $28 \times 28 \times 256$          |
| Conv / s1     | $1 \times 1 \times 256 \times 512$   | $14 \times 14 \times 256$          |
| 5x            | Conv dw / s1                         | $3 \times 3 \times 512 \text{ dw}$ |
|               | Conv / s1                            | $1 \times 1 \times 512 \times 512$ |
| Conv dw / s2  | $3 \times 3 \times 512 \text{ dw}$   | $14 \times 14 \times 512$          |
| Conv / s1     | $1 \times 1 \times 512 \times 1024$  | $7 \times 7 \times 512$            |
| Conv dw / s2  | $3 \times 3 \times 1024 \text{ dw}$  | $7 \times 7 \times 1024$           |
| Conv / s1     | $1 \times 1 \times 1024 \times 1024$ | $7 \times 7 \times 1024$           |
| Avg Pool / s1 | Pool $7 \times 7$                    | $7 \times 7 \times 1024$           |
| FC / s1       | $1024 \times 1000$                   | $1 \times 1 \times 1024$           |
| Softmax / s1  | Classifier                           | $1 \times 1 \times 1000$           |

**Figure 7:** Architecture of Mobilenet which consists of type of layer, filter size and input size

totype i.e. the centroid of the cluster is calculated and used as final representation.

Clustering is useful in number of application as it clusters the raw data and find out the hidden features in the database. So it is widely used in image classification, identifying fake news, spam filtering, image segmentation and so on.

A good example of clustering algorithm is the single linkage clustering. It is defined by fixing the value of a parameter  $\epsilon$ , two data point  $x, x'$  are in one cluster when  $d(x, x') \leq \epsilon$ .

## 2.6. TDA Algorithms

### Simplicial complexes

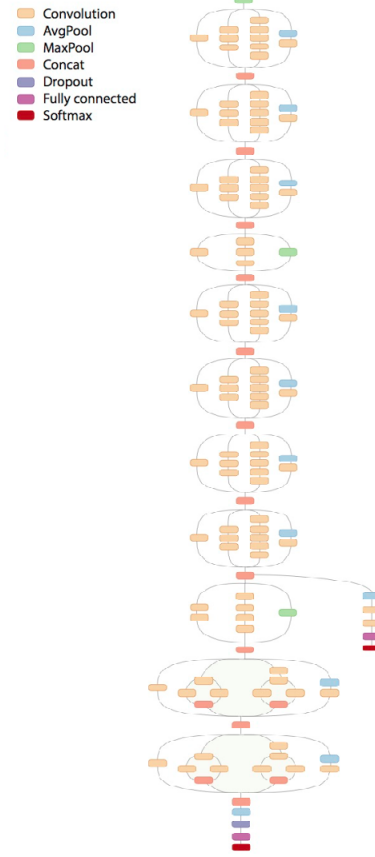
A classical way to represent discretized objects is using simplicial complexes, a collection of well-glued blocks called simplices. Formally, a  $k$ -simplex  $\sigma$  is the convex hull of  $k + 1$  affinely independent points. A 0-simplex is a single point, a 1-simplex is an edge, a 2-simplex is a triangle, a 3-simplex is a tetrahedron, and so on. Any simplex which is the convex hull of a nonempty subset of the points generating  $\sigma$  is called a face of  $\sigma$ . A simplicial complex  $\mathcal{K}$  is a finite set of simplices that each face of a simplex in  $\mathcal{K}$  belongs to  $\mathcal{K}$ , and each nonempty intersection of any two simplices in  $\mathcal{K}$  is a face of both.

### Persistent homology algorithms

Simplicial homology is a powerful tool in shape analysis, providing invariants for shape description and characterization. For a simplicial complex, it is possible to define some concepts like chain complex, filtration of a simplicial complex and homology groups that their ranks counts connected components, tunnels and holes of simplicial complex. Persistent homology allows for detecting the changes in the homology of a simplicial complex to detect changes in topological properties of simplicial complex with the help of filtration concept.

Summarizing the persistent homology method is as follows:

Inception v3

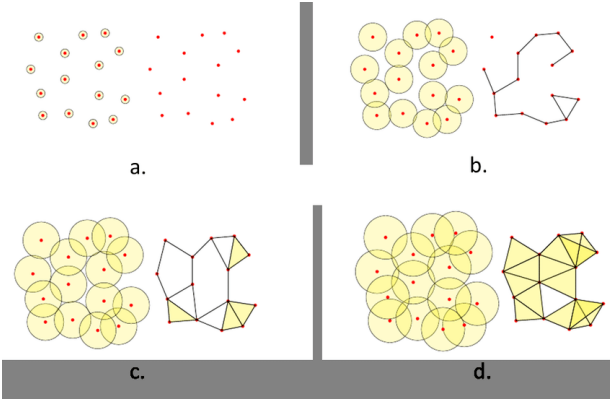


**Figure 8:** Architecture of Inception network for image classification

Let  $\mathbb{P}$  be a point cloud data. First we construct the Vietoris-Rips complex for  $\mathbb{P}$  as follows:

consider an increasing sequence of positive real numbers  $\epsilon_1 \leq \epsilon_2 \leq \epsilon_3 \leq \dots$ , then we construct a cover of circles with centers of points in  $\mathbb{P}$  and diameter  $\epsilon_1$ , so we have as many circles as the number of data points in point cloud data, next we draw an edge between the center of each two circle which have any intersection and therefore we have a simplicial complex  $VR(\epsilon_1)$ . We do the same process for all  $i = 1, 2, 3, \dots$ , as a result we have a filtration of complexes  $VR(\epsilon_i)$ . Reader can see one Vietoris-Rips complex constructed for a data set in figure 9 Now for analyzing the connection between points of dataset we compute different betti numbers of homological groups corresponding to filtration. These computations can be found in [15].

Since it is very hard to analyze the information about homological groups and holes we can use some visualization methods for detecting persistent homology like barcode, persistent diagram and landscape. A barcode represents each persistent generator(hole) with a horizontal line beginning at the first filtration level where it appears, and ending at the filtration level where it disappears, while a persistence diagram plots a point for each generator with its x-coordinate



**Figure 9:** Different Vietoris-Rips complexes constructed for a data set is shown in (a) to (d)

the birth time and its y-coordinate the death time. In other words persistent diagram can be explain as follows.

**Definition 1.** The  $p$ -persistent diagram  $D$  of a filtration

$$\emptyset \subseteq K_0 \subseteq K_1 \subseteq K_2 \subseteq \dots$$

is defined as follows.

Let  $\mu_p^{ij}$  be the number of independent  $p$ -dimensional classes that are Born in  $K_i$  and die entering  $K_j$ , then  $D$  is obtained by drawing a set of points  $(i, j)$  with multiplicity  $\mu_p^{ij}$ , where the diagonal is added with infinite multiplicity.

For comparing two persistent diagrams some metrics like bottleneck and wasserstein distances are defined.

**Definition 2.** Let  $D_1, D_2$  be two persistent diagrams and  $B$  be the set of all bijective functions  $\varphi : D_1 \rightarrow D_2$ . If  $\|\cdot\|_\infty$  be the supremum norm, then the bottleneck distance between two persistent diagrams  $D_1, D_2$  denoted by  $W_\infty(D_1, D_2)$  is defined as follows.

$$W_\infty(D_1, D_2) = \inf_{\varphi \in B} \sup_{x \in D_1} \|x - \varphi(x)\|_\infty$$

**Definition 3.** Let  $D_1, D_2$  be two persistent diagrams and  $B$  be the set of all bijective functions  $\varphi : D_1 \rightarrow D_2$ , then the Wasserstein distance between two persistent diagrams  $D_1, D_2$  denoted by  $W_p(D_1, D_2)$  is defined as follows.

$$W_p(D_1, D_2) = \left[ \inf_{\varphi \in B} \sum_{x \in D_1} \|x - \varphi(x)\|_\infty^p \right]^{\frac{1}{p}}$$

Reader can found more about persistent homology in [1].

### TDA Mapper method

Mapper is a tool from Topological Data Analysis (TDA) that provides a topological summary of the data. The Mapper algorithm was introduced by Singh, Mémoli and Carlsson [14] as a geometrical tool for analyzing and visualizing datasets.

**Table 1**  
Notation

| Symbol       | Explanation                                                                               |
|--------------|-------------------------------------------------------------------------------------------|
| $\mathbb{X}$ | Underlying space of point cloud data                                                      |
| $\mathbb{P}$ | Point cloud data ( $\mathbb{P} \subseteq \mathbb{X}$ )                                    |
| $\mathbb{Z}$ | Parameter space (usually $\mathbb{Z} = \mathbb{R}$ )                                      |
| $\mathbb{F}$ | Filter function ( $\mathbb{F} : \mathbb{P} \subseteq \mathbb{X} \rightarrow \mathbb{Z}$ ) |
| $\Gamma$     | Range of $\mathbb{F}$ restricted to $\mathbb{P}$                                          |
| $\mathbb{U}$ | A covering of $\mathbb{P}$                                                                |
| $\xi$        | Collection of subintervals of $\Gamma$ which overlap                                      |

Here we introduce a table of notations to explain the mathematics of mapper method.

Summarizing the mapper algorithm with respect to the above notations is as follows:

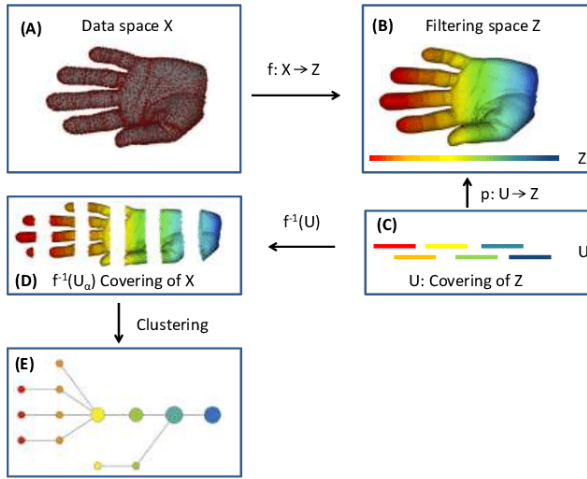
- First we start with a suitable filter function  $\mathbb{F} : \mathbb{P} \subseteq \mathbb{X} \rightarrow \mathbb{Z}$ ;
- Then we find the range of  $\mathbb{F}$  restricted to  $\mathbb{P}$  and call it  $\Gamma$ . Then partition  $\Gamma$  into subintervals  $\xi$  in order to create a covering of  $\mathbb{P}$  by inverse image  $\mathbb{F}^{-1}$  in the next step;
- For every subinterval  $\xi_i \in \xi$  we find the inverse image of  $\xi_i \in \xi$  under filter function and call it  $X_i$  that is the following set  $X_i = \{x \mid \mathbb{F}(x) \in \xi_i\}$ . The set  $\mathbb{U} = \{X_i\}$  form a covering for  $\mathbb{P}$  ( $\mathbb{P} \subseteq \bigcup_i X_i$ );
- For every element  $X_i$  of  $\mathbb{U}$  we cluster the points of  $X_i$  by single linkage clustering algorithm with a suitable metric, i.e for every  $X_i$  we have the set of clusters  $X_{ij}$ ;
- Every cluster  $X_{ij}$  would be represented as a vertex in simplicial complex where a family of vertexes  $\{X_{ij}\}$  spans a simplex if and only if the corresponding clusters have a point in common.

The intuitive idea behind Mapper is illustrated in figure 10 and can be explained as follows: suppose we have a point cloud data representing a shape, for example a hand. First we project the whole data  $X$  on a coordinate system with less dimension. in order to reduce complexity via dimensionality reduction (here we project the data on the hand to the parameter space  $Z$ ). Now we partition the parameter space into several bins with an overlapping percentage. Next, put data into overlapping bins. Afterwards, we use clustering algorithms in order to classify the points of each bin into several clusters. Once the previous stage is done, we can create our interactive graph.

## 3. Results

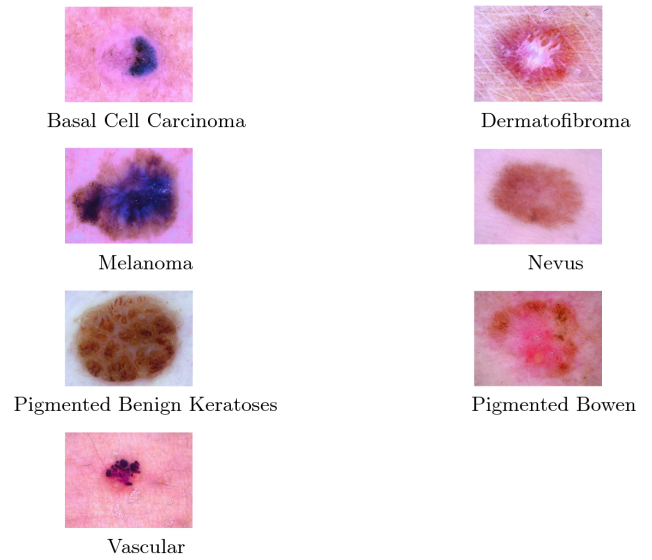
### 3.1. Data set

In this paper we have used the data set HAM10000. There are a total of 10015 dermatoscopic images of skin lesions labeled with their respective types of skin diseases. The images in the data set are separated into the following seven types of skin diseases:



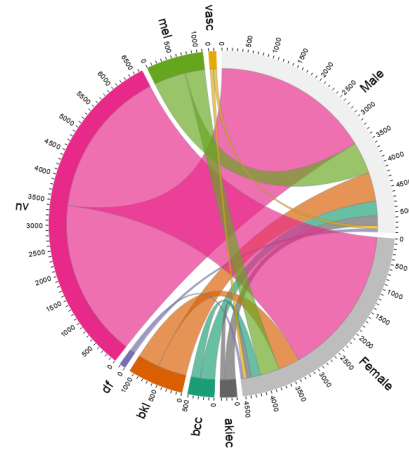
**Figure 10:** Mapper algorithm on hand shape data cloud: **B)** First we project the whole data cloud to parameter space. **C)** Then we partition the parameter space into overlapping bins (here showed as colored intervals). **D)** Then we find a cover of overlapping bins by computing inverse function of colored intervals. **E)** Next we use any clustering algorithm to cluster the points in the bins which each cluster would represent as a node of the graph and we draw an edge between two nodes if they share a common data point.

- Actinic keratosis is considered to be a noncancerous (benign) type of skin diseases. However, if left untreated, it usually develops into squamous cell carcinoma (SCC).
- Basal cell carcinoma is a cancerous type of skin lesion that develops in the basal cell layer located in the lower part of the epidermis. It is the most common type of skin cancer accounting for 80 percent of all cases.
- Benign keratosis is a noncancerous and slow-growing type of skin diseases. They can be left untreated as they are typically harmless.
- Dermatofibromas are also noncancerous and usually harmless, thus no treatment is required. It is commonly found pinkish in color and appears like a round bump.
- Melanoma is a type of malignant skin cancer that originated from melanocytes, cells that are responsible for the pigment of your skin.
- Melanocytic nevi are a benign type of melanocytic tumor. Patients with melanocytic nevi are considered to be at a higher risk of melanoma.
- Vascular lesions are composed of a wide range of skin lesion including cherry angiomas, angiokeratomas, and pyogenic granulomas. They are similarly characterized as being red or purple in color and are usually a raised bump.



**Figure 11:** Instances of Images of different lesions

Reader can see the instance image of these lesions in figure 11. Here we present some chord diagrams that show the weighted relation between different type of skin diseases and the other variables like gender of patient figure 12 or the location of lesion in patient figures 13 and 14.

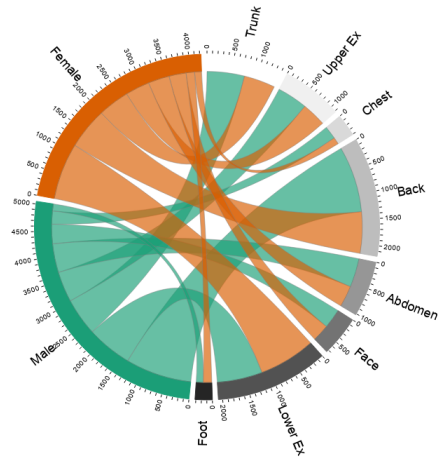


**Figure 12:** Chord diagram of the gender of patient and the type of lesion

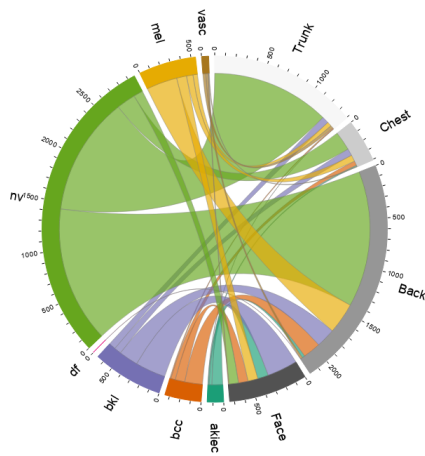
Also we use mosaic plot heatmaps to show the relationship between different variables like type of skin diseases, gender of patient and the location of lesion in patient figures 15, 16, 17.

### 3.2. TDA-Based Mapper Analysis online algorithm and our dataset

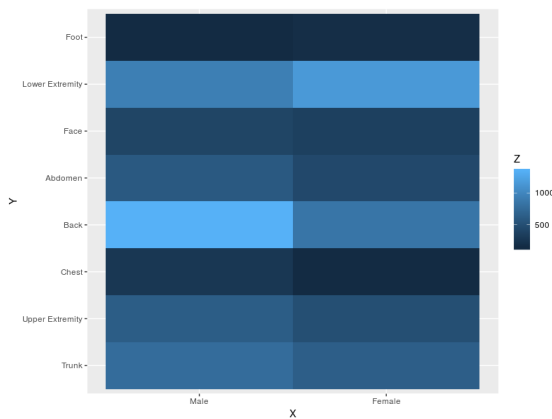
Implementation of mapper algorithm is already available in python packages like "Mapper" and "Kepler Mapper". In this article we used "Kepler Mapper" alongside other python packages. First we resize each image of our



**Figure 13:** Chord diagram of the gender of patient and the location of lesion

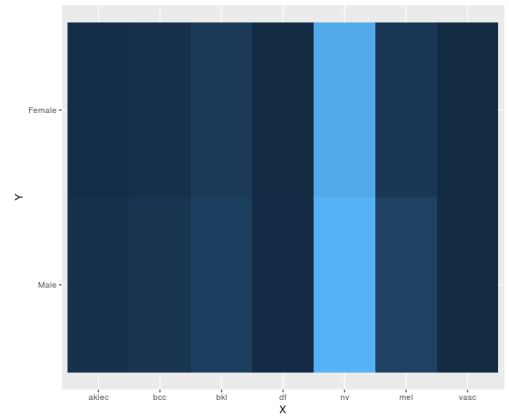


**Figure 14:** Chord diagram of the type and location of lesion

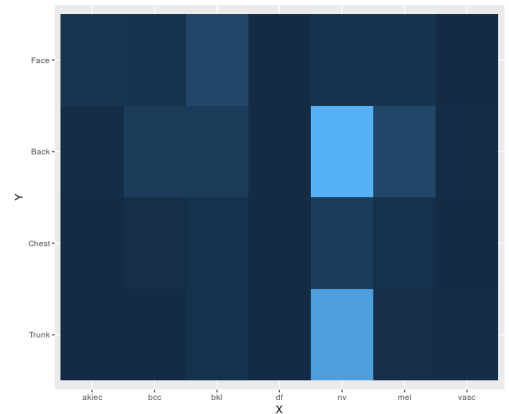


**Figure 15:** Heatmap of the relation between gender of patient and location of lesion

dataset into 28 by 28 pixel image, then we fed the dataset to a simple neural net, next we fed the output of neural net to



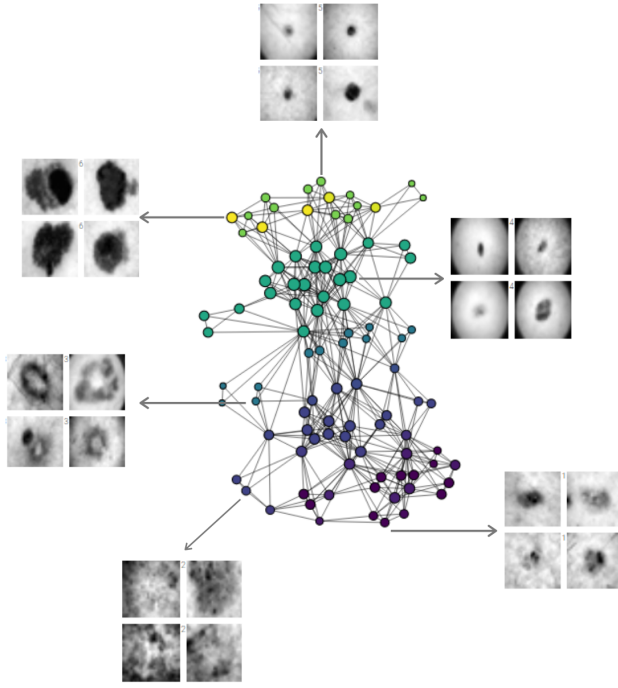
**Figure 16:** heatmap of the relation between gender of patient and type of lesion



**Figure 17:** Heatmap of the relation between type and location of lesion

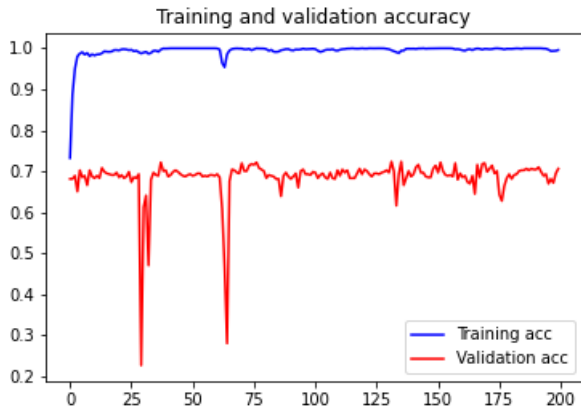
kepler mapper, for clustering we used “Agglomerative Clustering” available in “sklearn” package with “cosine” similarity and complete linkage. Finally mapper makes a simplicial complex for visualizing classification that has been shown in figure 18. To encapsulate the low-dimensional representation generated by the filtering step (neural net), Mapper employs binning (or partitioning), followed by partial clustering within each bin. The binning step partitions the low-dimensional space into overlapping bins by using two parameters: number of bins (or resolution ( $R=6$ )) and percentage of overlap between bins (or gain ( $G=0.4$ )). Within each bin, complete linkage clustering is performed to condense the time frames into a set of one or more clusters.

Since Mobilenet has depthwise convolution instead of regular convolution it is lighter and faster to train and it can be used for mobile applications. The stanford research in skin cancer with deep learning[16] has used Inception-V3 model to train their model, for computational shortages we used Mobilenet. We use the pretrained model and transfer learning in order to minimize the computational cost and training time. After 200 epochs the model reaches to stability in accuracy in comparison to Inception-V3 the result was acceptable with accuracy of over 70 percent in figure



**Figure 18:** Some sample images of nodes of simplicial complex induced from mapper

19 you can check the accuracy of our model in during 200 epochs.



**Figure 19:** Accuracy of mobilenet model on validation data during 200 epochs

| different epochs of layer 29 | Wasserstein distance  |
|------------------------------|-----------------------|
| epoch1 and epoch2            | 0.00729731097817421   |
| epoch2 and epoch5            | 0.007560431957244873  |
| epoch5 and epoch10           | 0.0027144961059093475 |
| epoch1 and epoch10           | 0.00823606550693512   |

**Table 2**

Wasserstein distances between different epochs of 29th layer.

| different epochs of last layer | Bottleneck distance   |
|--------------------------------|-----------------------|
| epoch1 and epoch2              | 0.004076704382896423  |
| epoch2 and epoch5              | 0.0047145746648311615 |
| epoch5 and epoch10             | 0.0047145746648311615 |
| epoch1 and epoch10             | 0.00508602149784565   |

**Table 3**

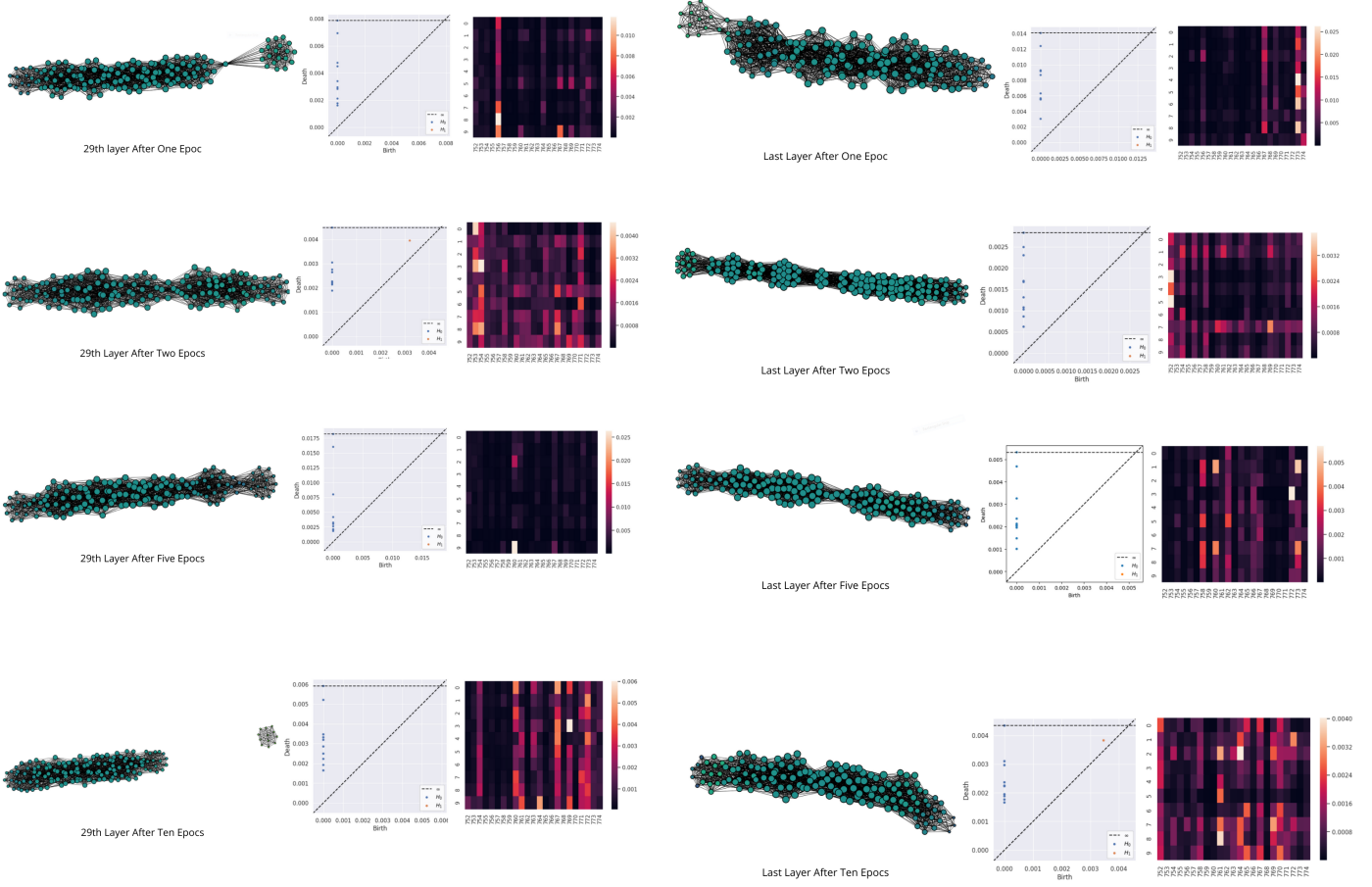
Bottleneck distances of different epochs of last layer.

### 3.3. Addressing black box problem

Convolutional neural networks are well adapted to image data. In this case, the input nodes are arranged in a square grid corresponding to the pixel array of Image. The nodes are composed in a collection of layers. A layer is called convolutional if it is made up of a collection of square grids identical to the input layer, and it is understood that the weights at the nodes in each such square grid involve only nodes in the previous layer that are very near to the corresponding node. Sometimes intermediate layers called pooling layers are introduced between convolutional layers, and in this case the higher convolutional layers are smaller square grids. MobileNet is a convolutional neural network that is trained on more than a million images from the ImageNet database. It can classify images into different categories. In fact the network has learned rich feature representations for a wide range of images. To train the neural network model of analysis, we used HAM10000 dataset with mobilenet neural network. To better understand the functionality of each layer during each epoch we used mapper and persistent to detect changes. Here we choose layer number 29 and the last layer of mobile net for epochs 1, 2, 5, 10 and then visualize the weights of them by both mapper and persistent diagram. The results of These calculation has been brought in figures20 and 21.

Mapper diagrams show that changes in weights of layer 29 in epoch 2 and 10 is very significant.

We bring the results of computing Wasserstein distances for different epochs of 29th layer in table 2 and the Bottleneck distances for different epochs of last layer in table 3 as follows.



**Figure 20:** Detecting changes in weights of layer 29 in different epochs

This demonstrates the capability of using topological data analysis to monitor and provide insight into the learning process of a neural network.

#### 4. Conclusion

In this manuscript we reviewed TDA algorithms like mapper and persistent homology and neural networks. Then for better visualization of classification problem with mobilenet neural net we used mapper algorithm alongside mobilenet algorithm on the dataset Ham10000. Also we visualized the black box problem of neural nets. The accuracy of this visualisation shows that classification problem can be done better by TDA algorithms alongside with neural nets.

#### Acknowledgments

The authors will thank the referees for reviewing this manuscript.

**Figure 21:** Detecting changes in weights of last layer in different epochs

#### References

##### Websites

- [1] 'KeplerMapper', <http://doi.org/10.5281/zenodo.1054444>, accessed Jan 2019

##### Journal articles

- [2] Aktas, M. E., Akbas, E., Fatmaoui, A. E.: 'Lecture Notes in Computer Science book series', Applied Network Science., 2019, 4, (61), pp. 1–28.
- [3] Buhrmester, V., Münch, D., Arens, M.: 'Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey', arXiv:1911.12116, 2019, pp. 249–274.
- [4] Carlsson, G.: 'Topology and data', Bull. Amer. Math. Soc., 2009, 46, pp. 255–308.
- [5] Edelsbrunner, H.: 'Persistence homology of networks: methods and applications', Persistent Homology In Image processing., 2013, 7877, pp. 182–183.
- [6] Gholizadeh, S., Seyeditabari, A., Zadrozny, W.: 'Topological Signature of 19th Century Novelists: Persis-

- tent Homology in Text Mining', big data and cognitive computing., 2018, 2, pp. 1–33.
- [7] Howard, A. G., et al.: ' MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications', arXiv:1704.04861, 2017.
  - [8] Lee, H., Kang, H., Chung, M. K., Kim, B., Lee, D.: 'Persistent Brain Network Homology From the Perspective of Dendrogram', IEEE TRANSACTIONS ON MEDICAL IMAGING., 2012, 31, pp. 2267–77.
  - [9] Lum, P. Y., Singh, G., Lehman, A., Ishkanov, T., Vejdemo-Johansson, M., Alagappan, M. Carlsson, J., Carlsson, G.: 'Extracting insights from the shape of complex data using topology', Sci. Rep., 2013, 3, pp. 2486–2495.
  - [10] Nielson, J. L., Paquette, J., Liu, A. W., Guandique, C. F., Tovar, C. A. et al.: 'Topological data analysis for discovery in preclinical spinal cord injury and traumatic brain injury', Nature Communications., 2015, 6, pp. 1–12.
  - [11] Sardi, M. E., Gilmore, J. M., Groppe, B., Florens, L., Washburn, M P.: 'Identification of Topological Network Modules in Perturbed Protein Interaction Networks', Scientific Reports., 2017, 7, pp. 1–13.
  - [12] Nicolau, M., Levine, A. J., Carlsson, G.: 'Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival', Scientific Reports., 2017, 7, pp. 1–13.
  - [13] Reiter, O., Rotemberg, V., Kose, K., Halpern, A. C.: 'Artificial Intelligence in Skin Cancer', Proc Natl Acad Sci U S A., 2011, 108, pp. 7265–70.
  - [14] Singh, G. Mémoli, F., Carlsson, G.: 'Topological methods for the analysis of high dimensional data sets and 3d object recognition', In Eurographics Symposium on Point-Based Graphics (eds Botsch, M., Pajarola, R.), 2007, 6, pp. 1–12.
  - [15] Zomorodian, A., Carlsson, G.: 'Computing Persistent Homology', Discrete and Computational Geometry., 2005, 33, pp. 249–274.
  - [16] Esteva, A., Kuprel, B., Novoa, R. et al. Dermatologist-level classification of skin cancer with deep neural networks. Nature 542, 115118 (2017). <https://doi.org/10.1038/nature21056>