E-ISSN: 2583-9667
Indexed Journal
Peer Reviewed Journal
https://multiresearchjournal.theviews.in

Received: 22-09-2023
Accepted: 30-10-2023

INTERNATIONAL JOURNAL OF ADVANCE RESEARCH IN MULTIDISCIPLINARY

Volume 1; Issue 2; 2023; Page No. 437-442

The role of TDA in improving the interpretability and explainability of neural network architectures and machine learning models

[1]Shivtirth Chaturvedi and [2]Manoj Sharma

[1]Research Scholar, PK University Shivpuri, Madhya Pradesh, India
[2]Department of Mathematics, PK University Shivpuri, Madhya Pradesh, India

DOI: https://doi.org/10.5281/zenodo.14599796

Corresponding Author: Shivtirth Chaturvedi

Abstract

More resistant to noise and able to retrieve information on higher dimensional aspects than what is initially seen in the data, topological invariants (shapes) of data may be studied within the overall framework of TDA. My presentation will focus on current and future efforts in order to apply persistent homology to the problem of intracellular network categorization. Data may be partitioned into hierarchical clusters using techniques from TDA, including persistent homology. In order to propose an updated version of the CNN and evaluate its performance in comparison to the original CNN, I investigate both theoretical and practical insights from topological clustering of graphs and pictures.

Keywords: TDA, neural network, architectures and machine learning models

Introduction

According to what was noted in one reason to build Convolutional Neural Networks (CNNs) was so that they might be scarified according to the geometry of the feature space. The two-dimensional array of characteristics, namely pixels, was the geometry used in convolutional neural networks for pictures. M. Robinson has also used the phrase topological signal processing to characterize the significance of topologies or geometries on feature spaces. We examine in this work a set of picture filters that is intimately associated with a Gabor filter subfamily whose shape is that of a famous geometrical object, the Klein bottle1. Not only may these filters enhance the data's feature set, but they can also be utilized to build convolutional neural network (CNN) analogues that perform better on several performance metrics. Additional scarification is achieved by adding layers that are derived from a graph-based discretization of the Klein bottle.

Through the application of extra structure on convolutional layers, we are able to apply the geometry of the Klein bottle and its related image filters. Neural networks that include these layers are referred to as Topological Convolutional Neural Networks (TCNNs). Image and video data are used in our investigations. According to the findings, TCNNs outperform traditional CNNs on a number of criteria.

When it comes to picture classification, deep neural network (NN) architectures are often considered the gold standard because to their exceptional accuracy and precision in differentiating between several classes. The most popular neural network design for picture categorization includes CNNs; for examples. A CNN's defining feature is the use of convolutional layers, which utilize weight sharing across slices and take advantage of the image's 2-dimensional topology to sparsity a fully connected network. The building blocks of convolutional neural networks (CNNs) are a network of linked layers to build more sophisticated global characteristics, such as object position and categorization, from more basic spatially local data, such as textures, lines, and edges. Video classification is another usage of CNNs; for examples. Convolutional neural networks (CNNs) have a number of significant limitations, such as models that are hard to understand, huge dataset requirements, and poor data generalization. Evidence suggests that CNNs' usefulness does not scale linearly with their size and complexity. This indicates that larger and more complex models will not be sufficient to sustainably improve picture categorization.

Literature Review

Topological deep learning computational challenge hosted

by the ICML 2023 Workshop on Topology and Geometry in Machine Learning is presented in this article. In order to participate in the competition, participants were invited to contribute to the TopoNetX (data processing) and TopoModelX (deep learning) python packages, which are topological neural network implementations grounded on literature. During the two months that the challenge was open, twenty-eight contributions were deemed qualified.
A wide variety of data analysis and machine learning applications make extensive use of discrete mathematical structures known as posets. Posets and data science have been the subject of continuous research for quite some time. This article provides a thorough literature overview of posets used in data analysis and machine learning, including the theory, algorithms, and practical implementations of these methods. There will also be an emphasis on the machine learning applications of applied lattice theory inside formal concept analysis.
Investigated the use of TDA for analyzing the architecture of neural networks. They used topological methods to study the loss landscapes of neural networks, revealing critical points and identifying structural patterns that influenced training efficiency. Their findings were pivotal in understanding the behavior of neural networks, providing insights into model optimization and generalization.

Rieck *et al.* (2019) [1]: Suggested a framework for detecting anomalies in high-dimensional data that is based on TDA. By leveraging persistence diagrams to identify deviations in the topological structure, the study demonstrated the effectiveness of TDA in detecting rare events. Applications in cybersecurity and financial fraud detection were highlighted, showing how TDA could outperform traditional statistical methods in identifying outliers.

Carrière *et al.* (2017): introduced kernel methods to improve persistent homology in machine learning's usability. They developed persistence kernels that allowed the seamless integration of TDA with assist vector machines and more kernel-based techniques. This approach facilitated applications in supervised learning tasks, including classification and regression, and proved particularly effective in biological data analysis.

Topological Convolutional Layers
For the purpose of 2D image categorization, we provide novel layers of neural networks that make up our TCNNs. For the purpose of video categorization, we provide new layers for TCNNs. shows how our structures are related to Gabor filters.

2D Images
In a typical convolutional neural network, (Cd,2) the correlation between cells in a two-dimensional image defines the L∐− spatial separation, which in turn determines locality, go to Definition. To this concept of locality, we provide new topological requirements in the form of metrics on topological manifolds. The following comment explains the overall method.

Remark: Every one of the layers specified in Definitions may be described by the following generic edge-defining correspondence C.
Let M be a manifold and let X, $X^r \subset M$ constitute two discretizations of M, denoting limited collections of points. Let $V_i = X \times Z^N$ and $V_{i+1} = X^r \times Z^N$ be successive layers in a FFNN. Fix a threshold $s \geq 0$. Let d be a metric on M. Define a correspondence $C(s) \subset X \times X^r$ by

$$C(s)^{-1}(\kappa^r) = \{\kappa \in X \mid d(\kappa, \kappa^r) \leq s\}$$

for all $\kappa^r \in X^r$. Together with another threshold $s^r \geq 0$, this defines a correspondence $C \subset V_i \times V_{i+1}$ by

$$C = C(s) \times C_{d,N}(s^r),$$

where $C_{d,N}(s^r)$ corresponds to Definition convolutional function. This indicates that

$$C^{-1}(\kappa^r, \underline{x}^r) = C_S(s)^{-1}(\kappa^r) \times C_{d,2}(s^r)^{-1}(\underline{x}^r)$$
$$= \{(\kappa, x) \in X \times Z^N \mid d_S(\kappa, \kappa^r) \leq s \text{ and } d_{ZN}(\underline{x}, \underline{x}^r) \leq s^r\} \text{ for all } (\kappa^r, \underline{x}^r) \in X \times Z^N.$$

First, we create a layer that, together with the standard methods, can locate itself based on its location on a circle. L∞− place inside a convolutional layer. Let $S^1 = \{\kappa \in R^2 \mid |\kappa| = 1\}$ constitute the plane's unit circle R^2. A typical discretization of S^1 is the set of n-th roots of unity $X = \{e^{2\pi ik/n} \mid 0 \leq k \leq n-1\}$ for some $n \geq 1$.

Definition: Let X, $X^r \subset S^1$ two separate representations of the circle. Let $V_i = X \times Z^2$ and $V_{i+1} = X^r \times Z^2$ to form an FFNN's consecutive layers. Set a limit $s \geq 0$.
The relationship between the rings $C_S(s) \subset X \times X^r$ is defined by

$$C_S(s)^{-1}(\kappa^r) = \{\kappa \in X \mid d_S(\kappa, \kappa^r) \leq s\}$$

for all $\kappa^r \in X^r$, when the measure dS is provided by
$d_S(\kappa, \kappa^r) = \cos^{-1}(\kappa \cdot \kappa^r)$ for $\kappa, \kappa^r \in S^1$.

If the shape of the correspondence that determines the edge $C \subset Vi \times Vi+1$ is the same for all other thresholds $sr \geq 0$, then we refer to Vi+1 as a circle one layer (COL).

$$C = C_S(s) \times C_{d,2}(s^r),$$

where $C_{d,2}(s^r)$ corresponds to Definition convolutional function. This indicates that

$$C^{-1}(\kappa^r, x^r, y^r) = C_S(s)^{-1}(\kappa^r) \times C_{d,2}(s^r)^{-1}(x^r, y^r)$$

$$= \{(\kappa, x, y) \in \mathcal{X} \times \mathbb{Z}^2 \mid d_S(\kappa, \kappa') \leq s \text{ and } d_{\mathbb{Z}^2}((x, y), (x', y')) \leq s'\}$$

for all $(\kappa^r, x^r, y^r) \in X \times Z^2$.

A layer based on a measure on Klein bottle K is then established to identify weights. The nodes and weights are shown graphically. Remember that by dividing R2 by the relations, the 2-dimensional manifold K is formed.
$(\theta_1, \theta_2) \sim (\theta_1 + 2k\pi, \theta_2 + 2l\pi)$ for k, l ∈ Z and $(\theta_1, \theta_2) \sim (\theta_1 + \pi, -\theta_2)$. The building incorporates an FK-squared embedding into the space of vectors expressing quadratic functions $[-1, 1]^2$inspired by the Klein bottle that was shown inserted and how it showed up in the CNN weights.

An integrated FK(θ1, Θ2) picture patch inside the Klein bottle being 'oriented' naturally by the angle θ_1. At an angle to the viewer's eyes, lines cut across the middle of the picture. $\theta_1 + \pi/2$; Observe the picture in the upper right corner. The embedded code is generated by

$F_K(\theta_1, \theta_2)(x, y) = \sin(\theta_2)(\cos(\theta_1)x + \sin(\theta_1)y) + \cos(\theta_2)Q(\cos(\theta_1)x + \sin(\theta_1)y)$,

Where
$Q(t) = 2t^2 - 1$. As given, F_K applies to the torus, with the two angles serving as its parameters. θ_1 and θ_2. Given that it meets, it does in fact define a function on K. $F_K(\theta_1, \theta_2) = F_K(\theta_1 + 2k\pi, \theta_2 + 2l\pi)$ and $F_K(\theta_1 + \pi, -\theta_2) = F_K(\theta_1, \theta_2)$.

2D IMAGES

Experiments and Results

The picture datasets provided are the basis for several investigations that we carry out. Working with separate datasets, we examine how Gaussian noise impacts TCNN training, TCNN activation interpretability, and TCNN learning rate as measured by testing accuracy over many batches. We study the generalizability tested TCNNs on many datasets, each trained on a different dataset. In each of these areas, we evaluate TCNNs in comparison to classic CNNs.

The picture datasets provided are the basis for several investigations that we carry out. Working with separate datasets, we examine how Gaussian noise impacts TCNN training, TCNN activation interpretability, and TCNN learning rate as measured by testing accuracy over many batches. We study the generalizability tested TCNNs on many datasets, each trained on a different dataset. In each of these areas, we evaluate TCNNs in comparison to classic CNNs.

Description of Data

We classify numbers on three datasets: USPS, MNIST, and SVHN. While all of these files include pictures of the numbers 0 through 9, their styles couldn't be more different. Specifically, it is quite easy for a human to tell which dataset a certain photograph is a part of. Because neural networks trained on one dataset tend to overfit to the specific style and idiosyncrasies of that dataset, they will have trouble generalizing to the other datasets, even if the numbers in each dataset seem different.

The datasets are available at three quite different resolutions: 162, 322, and 282. The datasets also differ greatly in size, with 7e3, 5e4, and 7e4 being the approximate values. In contrast to MNIST and USPS, which use handwritten numbers, SVHN uses typeset numbers. Numbers from SVHN are unprocessed images of typeset numbers that may include distortions, tilts, secondary digits, or other irregularities, as opposed to the direct two-dimensional imprints used by MNIST and USPS.

Additionally, we make use of the CIFAR-10 cat and dog labeled photos as well as the Cats vs. Dogs collection (which we refer to as Kaggle) of tagged images of cats and dogs. We downloaded a lot of empty or corrupted photos from the Kaggle Cats vs. Dogs dataset, so the size we're reporting here just represents the sum of all photos that can be loaded.

This seems to be the typical behavior for this dataset. One dataset has 2.5e4 photographs, whereas the other contains 1.2e4 images. Picture collection contained inside a single dataset have a resolution of 502 pixels and those in the other of 322 pixels. To ensure that the neural network trained on one dataset can be evaluated on both, we reduce the resolution of 322 rows of data from Kaggle so it matches CIFAR-10's dimensions. This allows us to assess the network's generalization accuracy.

Synthetic experiments

A key premise of this study is that constrained convolutional neural networks discusses convolutional neural networks (CNNs) for training on slices according to the Klein bottle topology and for using Klein bottle filters would immediately provide the model with very relevant local information. Thus, when global noise is introduced to the pictures, we anticipate using CF, KF, COL, and KOL layers to train a model would perform better than traditional CNNs. To put this theory to the test, we introduce class-correlated Gaussian noise. $N(\mu_k, \sigma_k^2)$ to our images, $\mathbf{z}_k = \mathbf{x}_k + \mathcal{E}_k, \mathcal{E}_k \sim N(\mu_k, \sigma_k^2),$ where k and $\mathcal{E}_k$ A key premise of this study is that constrained convolutional neural networks discusses convolutional neural networks (CNNs) for training on slices according to the Klein bottle topology and for using Klein bottle filters would immediately provide the model with very relevant local information. Thus, when global noise is introduced to the pictures, we anticipate that a model trained on our CF, KF, COL, and KOL layers would perform better than traditional CNNs. To put this theory before putting it to the test, we inject class-correlated typical random noise. of σ_k^2. the outcomes of this experiment conducted on MNIST. Training or testing pictures with Gaussian noise drastically reduces the performance of a traditional two-layer network. Given the presence of more to the training set, we expect the CNN to learn the class-correlated $\mu_k, \sigma_k^2,$, and then do badly on the non-Gaussian test set. include model loss, and it's evident that the 2-layer conventional CNN is accurately classifying inside the training set. We hypothesise that the TCNNs' exceptional success in the noisy training set experiment is a result of the fixed filters in the CF and KF layers. These filters compel a smoothing of the noise before the learnt layers, which in turn produces classifiers that are insensitive to global noise. For the noisy test set experiment, the TCNNs outperformed the competitors thanks to a comparable smoothing reasoning.

We will now choose parameters τ and ϋ ranging from 0 to.8 in increments of.2 and do the sampling $\mu_k \sim N(\tau, .04)$ and $\sigma^2 \sim \chi^2(1) \times \omega^2$. In the first column we simulate distributions $\mu_k \sim N(\tau, .04)$ and $\sigma^2 \sim \chi^2(1) \times .04$ evaluating the parameter τ for values between 0 and.8 with 0.2 increments.

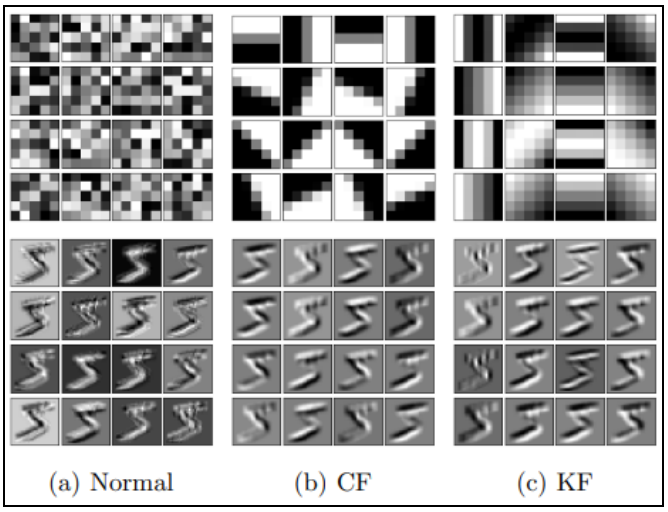

Fig 1: The table displays the activations and weights for three convolutional layers (a), (b), and (c) as they were tested on a handwritten 5 from MNIST.

Table 1: The picture datasets used in this study, including their sizes, dimensions, and origins.

Dataset	Size	Dimensions	Link
SVHN	60000	32^2	http://ufldl.stanford.edu/housenumbers
USPS	9298	16^2	https://web.stanford.edu/~hastie/StatLearnSparsity_files/DATA/zipcode.html
MNIST	70000	28^2	http://yann.lecun.com/exdb/mnist/
CIFAR-10	12000	32^2	https://www.cs.toronto.edu/~kriz/cifar.html
Kaggle	24946	64^2	https://www.kaggle.com/c/dogs-vs-cats

Distributions are modelled in the second column $\mu_k \sim N(.2, \omega^2)$ and $\sigma^2 \sim \chi^2(1) \times \omega^2$ testing ω from 0 to .8 in increments of .2.

In order to start, to see that we use data with Gaussian noise for training and actual data for testing, we look at the top row of simultaneously modifying τ and ϋ2. Despite the presence of an uncorrelated noise component, the class correlated signal of the extra noise is negligible, with μk being very small, since ω2 remains 0.04 when τ is 0. This noise lowers every model's precision. During the training process, the standard CNN is exceeded by the TCNN as τ grows. All of the models work in a similar way when ϋ2 is small, but when it's large, they all not do well enough. Nevertheless, the TCNN surpasses the CNN throughout a broad spectrum of ó2 values.

Interpretability

The variation in activations seen in images filtered using NOL, CF, and KF by MNIST., which allows us to compare the interpretability of CNNs and TCNNs. Training on the MNIST training data (n = 6e10) shows the activations which represent the results from the first layer of the CF, KF, and NOL model is used to experimentally generate the NOL filters. Eighteen CF filters represent sixteen equally interconnected angles on a sphere, in contrast to sixteen KF filters represent sixteen values for two Klein bottle angles, each with an equal spacing of four.

Rate of learning

Importantly, Networks that have a KF layer achieve high levels of accuracy more rapidly during training compared to those that do not, suggesting that they may be suitable for smaller datasets that need less training. Additionally, we demonstrate that the KF + KOL and KF + COL networks outperform NOL + NOL in terms of accuracy. Two datasets that benefited more from TCNNs than MNIST were the richer SVHN dataset and the lower-resolution USPS dataset. As a result, it seems that TCNN feature engineering works best when concealed or have weak local spatial priors.

Generalizability

We also assess the generalizability of models trained on dogs vs. cats' datasets from Kaggle and CIFAR, as well as models trained on MNIST and evaluated on SVHN. To see how testing accuracies changed throughout training. In the fields of numbers and cats vs. dogs, the TCNNs accomplish respectable generalization. When moving from MNIST to SVHN, the KF + KOL TCNN achieves a generalization accuracy of 30% and when moving from SVHN to MNIST, it achieves a generalization accuracy of over 60%. This is in stark contrast to the NOL + NOL traditional CNNs' 10% generalization accuracy between MNIST and SVHN, which is equivalent to pure guesswork. Keep in mind that these outcomes were unaffected by the inclusion of a pooling layer. The gap between TCNNs and CNNs for generalizing between the CIFAR cats vs. dog's dataset and the Kaggle dataset is less than it is for digit classification, but it is still noticeable. In this case, adding pooling layers improves TCNN generalizability even more but has little effect on CNN generalizability.

Gabor filters versus Klein bottle filters

It is possible to think of the Klein bottle filters that belong to the 5-parameter Gabor filter family given by equation (3.6) as a subset of the FK-embedded filters in equation (3.2), as previously shown. One may build a convolutional layer similar to the KF layer using the initialized and frozen filters for any subset of Gabor filters. Here, we evaluate a KF layer alongside one that is instantiated with a different set of Gabor filters and determined by the parameters listed in (3.7).

Although there is topological justification for the importance of the 2-parameter Klein bottle filter family in image analysis, the selection of a different Gabor filter family is currently not possible, hence, the family of two parameters in equation (3.7) is intrinsically somewhat arbitrary. We made this option based on a heuristic since these filters are high contrast and infrequently sampled from both φ and ω. In order to achieve strong contrast in the 3 × 3 filters, the empirically determined fixed parameters σ, λ, and γ are selected. In parameterization FK of Klein bottle filters, the variable parameter ϋ serves the same purpose as the parameter θ1 in terms of filter rotation.

$$\sigma = 2\pi,\ \lambda = \pi,\ \gamma = \frac{\pi}{8},\ \psi \in \left\{\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4}\right\},\ \omega \in \left\{\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4}\right\}.$$

By substituting these particular Gabor filters for the Klein filters in the KF topological layer, we are able to construct a 'Gabor' convolutional layer. Using the datasets examined in this study, we train a convolutional network that combines Gabor and NOL with a conventional NOL + NOL network and evaluate its performance in comparison. Take a look at 3.10. We discover that, in comparison to Gabor filters, Klein bottle filters work as well as, if not better than, the latter. To test the hypothesis, this indicates that the Klein bottle filters work well as Gabor filters in the topological layers of a TCNN. Because Gabor filters are members of a vast 5-parameter family, it is imperative that a very limited selection be selected for this sort of construction. The usually tiny kernels of CNNs make it much more challenging to understand a few of the characteristics.

Details of methods

Train/test splits

Concerning divides between training and testing, we do two types of experiments: (1) comparing the accuracy of TCNNs and CNNs on a specific dataset, and (2) training TCNNs and CNNs on a single dataset and validating it on another to assess the model's generalizability. As a first step, we created two sets of data: a training version and a test version. In the second kind, we train and test on all datasets. The train/test divides for Type (1) appear as follows:

For type (2), we use the lowest resolution in the comparison to determine the picture dimensions; in other words, we bring down the quality of the higher resolution photos so that we can assess generalizability more easily.

Meta parameter selection

It seems that we do not have a bias towards any single model in our experiment; rather, we choose metaparameters that enable us to compare the performance of TCNN with that of typical CNNs. To facilitate technique comparison, we stick to a standard, uncomplicated set of network requirements. All models must be able to properly traverse the optimized loss function before we can proceed with selecting an optimizer, batch size, learning rate, etc.

We uniformly apply a single configuration across all models in the tests. Of course, the specific outcomes change as a consequence of various configuration options, but our results are typically stable across all of them. Because our modifications stay inside the standard CNN architecture, we anticipate that meta-parameter selection won't significantly alter the relative merits of TCNNs compared to regular CNNs.

Detailed descriptions of the meta parameters and further information about our rationale are provided below. All of the models included in the publication have the same meta parameters within each figure. Various meta parameters are included in the following table according to the figure number. convolutional layers. Experiments not included were trained using a 2-layer convolutional neural network (CNN), which enables us to include a feature layer.

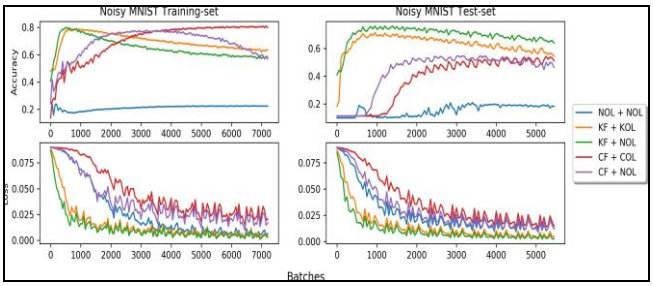

Fig 2: In order to verify the model's assumptions, two simulated experiments were conducted using noisy MNIST data. The results of the experiment are shown in the first column. The training data was exposed to Gaussian noise, but the testing data was not. Experiment results with original MNIST data used as training and damaged testing data due to Gaussian noise are shown in the second column. Accuracy in testing and training loss are shown in the first two rows, respectively.

Table 2: Datasets of numerical images: testing and trade

Dataset	Train	Test
MNIST	85%	15%
SVHN	80%	20%
USPS	80%	20%

Table 3: Associated metaparameters with the figures shown here.

Conv-layers	Conv-slices	Kernel size	LR	Batch size	Epochs
1	16	5	1e−4	100	1
2	64	3	1e−5	100	5
2	64	3	1e−4	100	1
2	64	3	1e−5	100	5

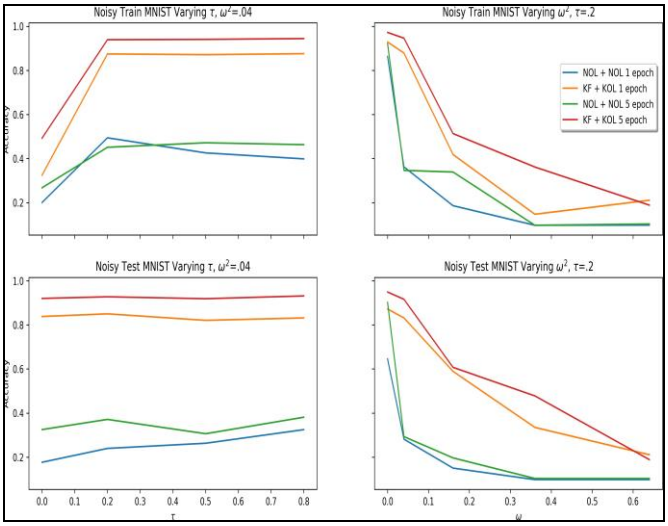

Fig 3: Examining the distributions of synthetic MNIST data with different means and variances for class-noise estimates. Both the first and second columns display the tested values of τ and ω2, respectively. Accuracy during training on a data set with Gaussian noise and testing on an unmodified test set are shown in the first row. Accuracy during training and testing on unmodified and noise-added data sets is shown in the second row. After 1 and 5 training epochs, we display the findings.

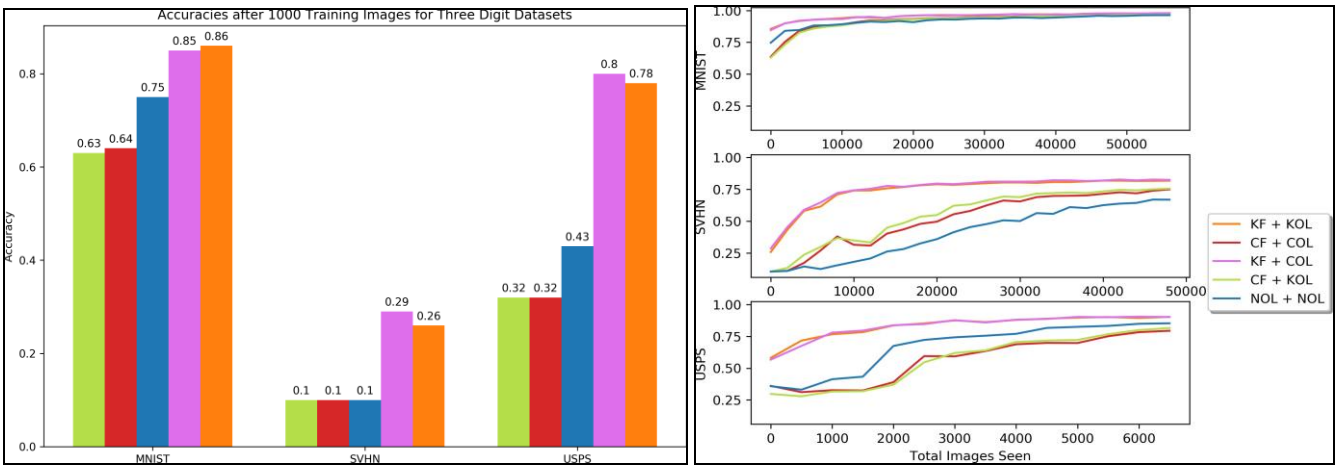

Fig 4: Testing accuracy comparisons after 1,000-image training (left). On the right, you can see a single epoch's worth of testing accuracy compared side by side. Training sets for MNIST, SVHN, and USPS included 60,000, 50,032, and 7291 pictures, respectively.

Conclusion

We evaluated various distance-based classifiers. A resampling strategy, a persistence diagram vectorization, and a support vector machine classifier make up our most effective method Theoretical understanding of the space of natural pictures and results on the long-run topologies of well-trained CNNs are the building blocks of this framework. Several such formulations were compared and contrasted with one another and a standard CNN in terms of their empirical findings. Based on actual facts and/or theoretical linkages to a logical manifold, this work may be used as a template to construct endless models with expanded topologies. In a number of tests involving the categorization of images and videos, I proved that this generalization was effective.

References

1. Rieck B, Sadlo F, Leitte H. Topological machine learning with persistence indicator functions. IEEE Trans Vis Comput Graph. 2019;26(1):113–123. https://doi.org/10.1109/TVCG.2019.2934701.
2. Wasserman L. Topological data analysis. Annu Rev Stat Appl. 2016;5:1–27. https://doi.org/10.1146/annurev-statistics-031017-100045.
3. Smith A, Dlotko P, Zavala V. Topological data analysis: Concepts, computation, and applications in chemical engineering. Comput Chem Eng. 2020;146:107202. https://doi.org/10.1016/j.compchemeng.2020.107202.
4. Park S, Hwang Y, Yang BJ. Unsupervised learning of topological phase diagram using topological data analysis. Phys Rev B. 2022;105. https://doi.org/10.1103/PhysRevB.105.195115.
5. Skaf Y, Laubenbacher R. Topological data analysis in biomedicine: A review. J Biomed Inform. 2022;130:104082. https://doi.org/10.1016/j.jbi.2022.104082.
6. Dey T, Wang Y. Computational topology for data analysis. 2022. https://doi.org/10.1017/9781009099950.