EPiC Series in Computing
Volume 83, 2022, Pages 86–96
Proceedings of 14th International Conference on Bioinformatics and Computational Biology

Interpretable Image Classification Model Using Formal Concept Analysis Based Classifier

Minal Khatri[1], Adam Voshall[1], Surinder K. Batra[2], Sukhwinder Kaur[2], and Jitender S. Deogun[1]

[1] University of Nebraska-Lincoln, Lincoln, NE, USA
minal.khatri@huskers.unl.edu,, voshalla@gmail.com, deogun@cse.unl.edu
[2] University of Nebraska Medical Center, Omaha, NE, USA
sbatra@unmc.edu, skaur@unmc.edu

Abstract

Massive amounts of data gathered over the last decade have contributed significantly to the applicability of deep neural networks. Deep learning is a good technique to process huge amounts of data because they get better as we feed more data into them. However, in the existing literature, a deep neural classifier is often treated as a "black box" technique because the process is not transparent and the researchers cannot gain information about how the input is associated to the output. In many domains like medicine, interpretability is very critical because of the nature of the application. Our research focuses on adding interpretability to the black box by integrating Formal Concept Analysis (FCA) into the image classification pipeline and convert it into a glass box. Our proposed approach produces a low dimensional feature vector for an image dataset using autoencoder followed by a supervised fine-tuning of features using a deep neural classifier and Linear Discriminant Analysis (LDA). The low dimensional feature vector produced is then processed by FCA based classifier. The FCA framework helps us develop a glass box classifier from which the relationship between the target class and the low dimensional feature set can be derived. Further, it helps the researchers to understand the classification task and refine it. We use the MNIST dataset to test the interfacing between deep neural networks and the FCA classifier. The classifier achieves an accuracy of 98.7% for binary classification and 97.38% for multi-class classification. We compare the performance of the proposed classifier with Convolutional neural networks (CNN) and Random forest.

1 Introduction

The advancements in digital technology have substantially contributed to the availability of histopathological image datasets. Image classification is a widely used tool in the health care. Automating image classification has the potential for early and accurate diagnosis, helping save lives and reduce the overall cost of health care. Many state-of-the-art techniques use Convolutional neural networks (CNN's) for image classification problems [19]. The most significant advantage of using CNN is that it can automatically learn features from image data and handle non-linearity in the data. With the development of deep architectures for complex data, CNN's involve computation and optimization of millions of parameters. As an example, the network VGG-16 [18] designed for ImageNet [4] incorporates about 138 million parameters, which are optimized over millions of images. The VGG-16 network is often used to transfer learn on small medical image datasets and has shown promising results [16].

In the current literature, CNN's are often treated as a black box technique. CNN's approximate a function representing the data distribution, but the parameters learned in the architecture do not give any insight into the function approximated [2]. For example, consider applying a CNN in health care to classify histopathological images as "cancerous" or "benign." From a mathematical perspective, the neural network will build an approximate function $f(C) = R$ with an error rate acceptable to the domain. However, the function f is not returned; can be arbitrarily complex; and might change with a change in input data. As a result, it is difficult to interpret the output when CNN and a domain expert do not come to the same conclusion. The interpretability is critical to high-stake decision-making domains like health care.

Previous studies have hypothesized the internal working of a CNN classifier in two steps: (1) feature extraction and (2) classification [7]. The first few layers learn a set of features that amplify the aspects of input that are important for discrimination and suppressing irrelevant variations. In image data, for example, learned features would represent the presence or absence of edges at particular orientations and locations in the image. The subsequent layers assemble these features into a larger combination that corresponds to complex shapes. The final layers would detect input images as combinations of these shapes and classify them in one of the given classes. The above interpretation is not sufficient to understand how different features are related to different classes in the dataset.

The objective of this study is to develop a domain independent image classifier that can be used to interpret the classification task. We propose a feature extraction framework based on deep neural networks and Linear Discriminant Analysis (LDA) [9] to reduce the number of features to only those relevant to the classification of an image. In the next step, we use Formal Concept Analysis [23] framework to build an interpretable classifier. FCA forms hierarchical groupings of data using the low dimensional feature vector, where each grouping is defined as a "concept"; set of objects that share common attributes. Once the data is grouped, we can identify the relationship between the features and the target class, enhancing the interpretability of the classification output. Classifiers based on Formal Concept Analysis have several advantages over CNN's as discussed below. Formal Concept Analysis groups the data into meaningful concepts which is a unit of thought in humans. Experts in the domain can visualize the groupings of data to understand the relationship between the feature vector and the target class. Additionally, the Formal Concept Analysis based classifier can handle the change in input data without the need for retraining. An addition of features or objects in the input dataset would either fit into an existing concept or a new concept is added. Finally, the FCA framework gives insight into the internal logic of the classification task and can be modified incrementally.

The main objective of this study is to contribute towards the development of a novel approach for image classification, which adds interpretability to the classification task. In our approach, we use neural networks for feature extraction and an FCA based classifier to add transparency to our model and interpretability to the output. For the purpose of this study, we use the MNIST dataset [13]. In the future, we plan to apply the proposed approach to medical images, such as histopathological images.

The paper is organized into sections: the following section briefly describes the feature extraction techniques and formal concept analysis framework, section 3 describes the proposed classifier, the experiments and results are summarized in section 4, and finally, the conclusions and scope of future work are discussed.

2 Background

In this section, we describe the feature extraction techniques and the Formal Concept Analysis framework.

2.1 Feature Extraction

A large dimensional dataset makes most of the classification methods infeasible due to high computational complexity. Therefore, feature extraction techniques are used to reduce the dimensionality of large datasets. Feature extraction transforms data from high dimensional space to a relatively low-dimensional space by identifying important features and eliminating noise and redundancy. It allows the application of classification methods to the low dimensional representation instead of the entire feature set. The feature extraction methods vary from linear transformation methods like Principal Component Analysis (PCA) [11], Linear Discriminant Analysis (LDA) [9] to non-linear transformation methods like autoencoders [17]. The recent interest in deep learning has fueled the popularity of Autoencoder(AE) as powerful unsupervised feature extraction.

Deep autoencoders have been applied successfully as a dimensionality reduction technique in many fields, such as the health sector[15], telecom[1], speech recognition [14] because it can automatically learn features from images, audio, and text data.

2.2 Formal Concept Analysis (FCA)

Formal concept analysis is a branch of mathematical lattice theory introduced in the early 1980s by Rudolf Wille [21]. FCA is a framework used to structure, analyze and visualize data and its relationships [6]. It is based on the notion of ”concept” and ”lattice.” A concept is an abstraction of human thought, allowing meaningful groupings of objects that share common attributes [22]. Moreover, the lattice is a visual representation used to analyze hierarchies of these groupings to find patterns, similarities, and exceptions.

The input to Formal Concept Analysis is a formal context or incidence table. The rows of the table represent a set of objects G, and the columns represent a set of attributes M. A binary relation $I \subseteq G \times M$ exists between objects and attributes. Formally, the context is a triple $K = (G, M, I)$. FCA generates a set of concepts from the given formal context. A concept is a maximal collection of objects that possess common attributes. More formally, a concept C is a pair of sets (A, B) such that:

$$A = \{a \in G | \forall b \in B : (a, b) \in I\}$$

$$B = \{b \in M | \forall a \in A : (a, b) \in I\}$$

where A is considered the extent of the concept, and B is the intent of the concept. Further, the concept lattice represents the hierarchy of all the concepts of a given context. It is represented by a line diagram called ”Hasse diagram” [23]. Concepts can have relationships with many other concepts; some concepts are more general than others. The relationship of generalization and specialization is modeled using the subconcept-super concept relation. The extent of the subconcept is a subset of the extent of the super concept. This is equivalent to the intent of the super concept is a subset of the intent of the sub-concept [20].

Previously, FCA was majorly used in the Information Retrieval field for formalizing concepts and conceptual thinking. In the recent works, FCA is often used for clustering [3], association rule mining [24], and determining classification rules [8].

3 Method

The proposed classifier consists of two main components: (1) Autoencoder and Linear Discriminant Analysis (LDA) for feature extraction; and (2) Formal Concept Analysis based classifier.

3.1 Autoencoder and Linear Discriminant Analysis (LDA) for Feature Extraction

Feature Extraction is used as a pre-processing step in classification problems to avoid the curse of dimensionality. In our approach, we use FCA as a classifier which creates 2^n concepts in the worst case, where n is the number of features. To make the problem feasible, we need to reduce the number of features to a more manageable number.

We propose a three-step feature extraction framework represented in Figure 1. In the first step, we use autoencoders to obtain a lower-dimensional representation of input data in an unsupervised manner (i.e., it does not consider the class labels). The autoencoder is trained over a number of iterations to minimize the reconstruction loss. The main advantage of using autoencoders is that they can significantly reduce the number of features while retaining the features that are important for the reconstruction of the input. In the second step, we replace the decoder with a dense layer classifier to perform classification. The supervised training of encoder will help us to obtain features that achieve higher performance on classification. The weight of the encoder is initialized with the weights obtained after training in the first step. The model is trained over a number of iterations which helps in minimizing the loss function. We obtain the output of the encoder as the new low dimensional feature set. In the third and final step, we use the new low dimensional feature set as input for the Linear Discriminant analysis function. The LDA function projects the data on a new linear axis, which maximizes the distance between the means of each class in the data and minimizes the variation within each class. Consider the two dimensional dataset shown in Figure 1 where each color represents a different class. The LDA function uses the information from both features and creates a new feature that maximizes the separation between two classes. The main advantage of using LDA is that it can further reduce the number of features by maximizing separability between classes which helps in making better decisions in the classification task. The low dimensional feature vector obtained from LDA is used in the classification task. The specific details for each component of our classifier is presented in the next section.

3.2 Formal Concept Analysis (FCA) Classifier

In our proposed method, we are using Formal Concept Analysis as a classifier. The classification approach based on Formal Concept Analysis allows us to extract the relationship between features and the different class labels based on the concepts discovered from data. Conceptually speaking, our classification method consists of a learning step and a classification step. In the learning step, we organize the information extracted from a training set in the form of concept lattice where we define a class label for each concept. In the classification step, we predict the class of instances in the testing set based on concepts in the training lattice.

The input to FCA based classifier is a formal context; an incidence table where rows represent the set of images and the columns represent the binary features obtained after the feature extraction. In addition, because we need to transform the output of LDA (multi-valued features) into a formal context (binary features), as a pre-processing step, we first normalize the output of LDA and then set a threshold to transform multi-valued features into binary features. All

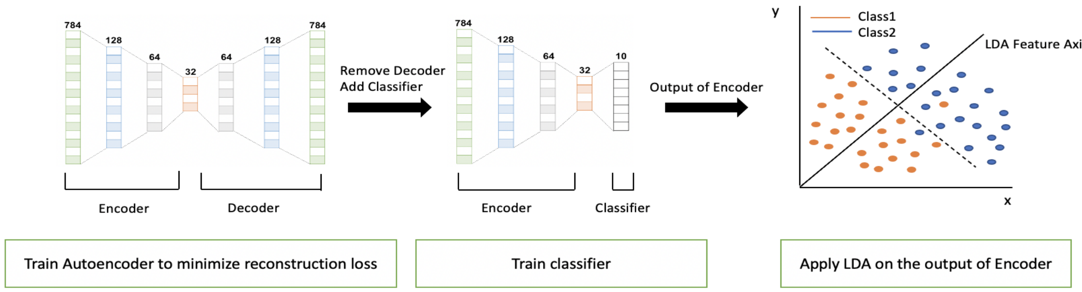

Figure 1: Feature Extraction Framework

the values above the threshold are assigned as True and the remaining values as False. Note that the same threshold is used for all features in both training and testing data sets.

We implemented the FCA classifier algorithm in Python and the concepts package is used to construct the concept lattice. There are three steps in the algorithm as described below:

1. Constructing training lattice: In the first step, we construct the training lattice using the training set's formal context. To build the lattice, we do not need all the images from the training set because images with the same feature vector will not add any new concept to the lattice. Therefore, to speed up the time for construction of lattice, we need images that have unique feature vectors forming a unique feature set X_t. However, we need to keep track of the class labels of images with same feature vector. We define a new class label set, y_t corresponding to set X_t. Each entry in y_t is a list of class labels of all the images that have the same feature vector in X_t. For example, the data contains two images with feature vector $[1, 1, 0, 0]$, $[1, 1, 0, 0]$ belonging to class 0 and 2 respectively. We add $[1, 1, 0, 0]$ to the set X_t and $\{0, 2\}$ to the set y_t. As a result for this step, a training lattice is constructed from the unique training feature set X_t.

2. Defining a class of a concept in training lattice: Once the training lattice is constructed, we define the class label for each concept in the lattice. The class label with a maximum number of images in the extent of the concept is assigned as the concept class. We break ties arbitrarily. Assigning a class label to each concept helps to identify the relationship between the feature set (concept's intent) and class labels (concept's class).

3. Classify the testing set: In the final step, we test the developed classifier on the testing set's formal context. The features are binary, therefore, we choose hamming distance to find the similarity between the two feature vectors. To obtain the predicted class label for each image in the testing set, we calculate the hamming distance of the feature vector with every concept's intent in the training lattice. The concept with the smallest hamming distance is chosen because the testing image is more similar to the images belonging to this concept in the training lattice. The corresponding concept class is assigned as the predicted class label for the testing image. Note, we break the ties arbitrarily.

We use "accuracy" to evaluate the performance of our proposed classifier. Moreover, the lattice diagram helps in understanding the hierarchical learning and relationship between features and class labels. The details of the parameters varied for each component and the corresponding results are presented in the next section.

4 Experiment and Result

We tested our approach on the MNIST dataset. The MNIST dataset [13] is a handwritten digit image dataset that consists of 60,000 images in the training set and 10,000 images in the testing set. Each image is a 28×28 grayscale pixels and each image belongs to one of the ten classes representing integer values from 0 to 9. The programming language used for implementation is Python. All the programs were run on Intel Xeon processor with 16 cores and 64GB RAM. The runtime of the FCA algorithm depends exponentially on the number of features n. Therefore, we will not perform a parameter search on the value of n in our study. In this section, we briefly discuss the parameters varied for each component and the corresponding results.

4.1 Autoencoder and Linear Discriminant Analysis (LDA)

In the first step of feature extraction framework, we tested different architectures of dense layer autoencoder and obtain the low dimensional feature sets. The implementation uses the Keras framework in Python. The network takes a flattened input image with size 784 features. We used *Relu* as an activation function for each layer with early stopping for all the architectures. The error function used to train the autoencoder is the *mean squared error* and is trained for 50 epochs. After training, we remove the decoder and add a dense layer with 10 neurons for 10 class classification. The categorical cross-entropy loss function is minimized. We report the results from the three best architectures in Table 1, where Encoding represents the number of features returned by the encoder.

Architecture	Encoding	Cross_entropy
784, 256, 128, 64, 32, 64, 128, 784	32	0.02
784, 256, 128, 64, 32, 16, 32, 64, 128, 256, 784	16	0.02
784, 512, 256, 128, 64, 32, 64, 128, 256, 512, 784	32	0.01

Table 1: Results of Autoencoder on MNIST

In the next step, we use the *LinearDiscriminantAnalysis* function in sciket-learn to implement LDA. The LDA function takes an argument *n_components* to determine the number of features to be returned. The value of *n_components* can be refined based on experimentation and it ranges from 1 to [number of classes -1]. For MNIST dataset (10 classes), we experiment with values in the range [4,9]. The number of features returned should be atleast 4 because the FCA classifier would need at least 2^4 concepts to represent 10 classes (2^3<10). In the experiments, we observed that the classification accuracy increases as the number of features increases from 4 to 9. Therefore, we report the results for two values of *n_components*, which is 4 and 9. After applying LDA, for each architecture, we obtain two feature sets. One has 4 features and the other has 9 features. The average runtime for feature extraction is approximately 15 minutes for each architecture.

4.2 Formal Concept Analysis (FCA) Classifier

In the second step, we perform the classfication using the FCA classifier which takes binary features as input. We transform the LDA output (multi value features) to formal context (binary features) by using a threshold. The output from LDA is normalized and then the the value of threshold vary between 0 to 1 with an increment of 0.001. Therefore, we obtain a total

Architecture	n_components	Accuracy% (Multi Class)	Accuracy% (Binary)
1	4	63.98	87.4
1	9	92.58	97.83
2	4	50.68	86
2	9	95.74	97.5
3	4	59.35	80.5
3	9	97.38	98.7

Table 2: Results of Multi-class and Binary Classification on MNIST

of 1000 feature set for each architecture. The FCA classifier is executed for each feature set and the average runtime is about 6 hours. We perform multi-class and binary classification for all of the feature sets. In addition, we compare the results with CNN and Random forest classifier. Further, we discuss the lattice representation and incremental behavior of FCA classifier.

Classification: We perform a ten class classification and the results are reported in Table 2. As we can see from Table 2, for each architecture the multi class accuracy is increased by more than 25% when the number of features returned by *n_components* changes from 4 to 9. To perform binary classification, we transform the ten classes into two classes. For simplicity, the first five classes $[0, 4]$ are assigned to label 0 and the remaining classes $[5, 9]$ were assigned to label 1. The results are reported in Table 2. The highest accuracy reported is 97.38% for multi class and 98.7% for binary classification for the architecture 3 when the feature set is of size 9. Further, binary classification gives a higher accuracy than multi-class classification which suggest that there is similarity between objects within classes [0,4] and [5,9].

Lattice representation: Now, we briefly discuss the usefulness of concept lattice representation to identify the relationship between low dimensional feature set and the class label. As an example, a training lattice corresponding to architecture 1 with 4 features that achieves an accuracy of 63.98% is presented in Figure 2. In the concept lattice, the nodes represent the concepts labelled as $\{C1, C2, ...C16\}$ and the labels $\{a, b, c, d\}$ represent the features and the labels $= \{1, 3, 4, 5, 6, 7, 9\}$ represents the concept class. The intent and extent of each concept can be read from the line diagram by following the simple reading rule: the intent of a concept is the set of attributes on all the paths leading in the upward direction from the node and extent is the set of objects on all the paths leading in the downward direction from the node [22]. Due to space limitations, we do not represent the extent.

Each concept (or node) in the lattice in Figure 2 represents a relationship between the intent and class label of the concept. According to concept $C6$, if an object has value of features c and a above threshold then it is classified as an object of class 9. In addition, the lattice can be used to analyze the relationship between different classes. For example: The concept C4 with intent $\{b\}$, concept class 3 and the concept C5 with intent $\{d\}$, concept class 6 are superconcepts of the subconcept C11 with intent $\{b, d\}$, concept class 5. This implies that both the concept classes classes 3 and 6 shares similarity with concept class 5. It is interesting to note that the upper half portion of digit 6 and lower half portion of digit 3 overalps with digit 5. This is an example of hierarchical learning that we expect the FCA classifier to learn. In addition, we observe that the classes 0 and 2 are not represented by any concept in the lattice. Therefore, nearly 20% images are incorrectly classified. The classes are not discovered in the lattice because it requires more features to distinguish them. Therefore, as we increase the number of features to 9 for architecture 3, the accuracy increased to 92.58 (Table 2).

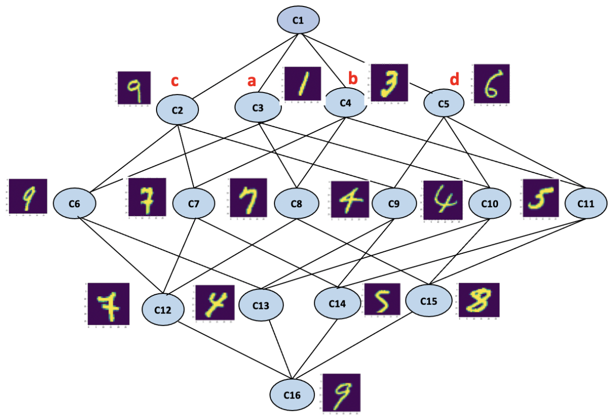

Figure 2: Concept Lattice

To understand the incremental behavior of FCA from the lattice in Figure 2, we construct a lattice with three feature a, b and c. This lattice will not have the concepts [C5, C9, C10, C11, C13, C14, C15, C16]. Now, if we add feature d, it will add new concepts to the lattice without changing the lattice structure obtained so far. Further, to add new data object to the lattice if the intent of an object is a proper subset if an existing concept; the object the added to the extent of the object. In case the intent is not matched with any concept we introduce a new concept. Therefore, lattice can be updated with new information by either adding a new concept or extending the intent/ extent of an existing concept.

Comparison of Results: We compare the performance of the proposed FCA classifier with CNN and Random forest classifiers on ten class classification. The results are reported in Table 3. In the current literature, the highest accuracy reported for CNN is 99.8% [10] and random forest is 96.2% [5] on the MNIST dataset. Note that our proposed FCA classifier obtains the highest accuracy, which is 97.38%. Although CNN achieves a little higher accuracy on the MNIST dataset, FCA has several advantages over CNN. Firstly, FCA is a glass box and present interpretability to the classification task. Moreover, FCA facilitates examination of classification hierarchy by providing a visual representation of data in the form of a lattice. Secondly, during the classification, data can be added and removed from the lattice without the need for retraining. Further, the FCA classifier outperforms the Random forest classifier, which is a human interpretable model. Random forest[12] is a widely used machine learning algorithm that solves classification problems. It uses an ensemble of decision trees where each decision tree is trained on a random sample of the training dataset. A slight change in the input data can cause a significant difference in the decision tree structure due to random sampling of training data. Unlike random forest, the Formal Concept Analysis based classifier can handle the change in the input dataset. FCA is an incremental model that can incorporate the addition of features or objects into the existing lattice.

Our proposed image classifier is a generic framework which can be applied to all image datasets. However, for different datasets, we can modify the feature extraction framework by incorporating new techniques that improve the classification result. In this paper, the proposed classifier gives promising results on the MNIST dataset. We will transition to apply the proposed model on medical images in future work. The medical image datasets are usually small because the procedure of collection of data is expensive. The limited availability of data impacts CNN's

Random Forest%	FCA %	CNN%
96.21	97.38	99.8

Table 3: Comparison of Accuracy

performance because, for high-dimensional images, CNN would require a large amount of data to discover the patterns within the data. However, it would have less impact on our proposed image classifier because we can train the feature extraction framework on a broad range of histology images and then train the FCA on the subsets of images of specific cancer types. This is a foundational work in the domain and it can lead to several future research directions.

5 Conclusion and Future Work

Convolutional neural networks have been applied successfully in many domains for image classification. However, it is treated as a black box technique and interpretability is important in domains like medicine, where deep learning models are used to make critical decisions. This study aims to develop an understanding of the classification task by integerating a Formal concept analysis framework into the image classification pipeline. While the FCA classifier has an exponential runtime ($O(2^n)$, n is the number of features) in the worst case, we can overcome this limitation by controlling the number of features obtained from the proposed feature extraction framework which utilize the power of deep autoencoders for feature extraction, followed by a supervised fine-tuning of features using a deep classifier and LDA. The FCA based classifier has two main advantages over state-of-the-art CNNs for image classification. Firstly, we can interpret the classification output using the lattice representation. Secondly, FCA classifier can add data at any point in the training without retraining the network. The proposed classifier achieves higher accuracy (98.7%) for binary classification compared to accuracy (97.38%) for a multi class classification and outperforms the Random forest model. For future work, we will incorporate a feedback learning mechanism into the FCA classifier to improve the classification results. The feedback mechanism was not implemented in this study because of the presence of LDA between the FCA classifier and autoencoder. We will also consider different thresholds instead of a global threshold for the feature set. Finally, this research's main objective is to apply this interpretable model in the health domain for colon cancer image classification.

6 Acknowledgments

This work is supported in part by Small Business Innovation Research (or SBIR) program (Grant No. R43 235984).

References

[1] Maha Alkhayrat, Mohamad Aljnidi, and Kadan Aljoumaa. A comparative dimensionality reduction study in telecom customer segmentation using deep learning and pca. *Journal of Big Data*, 7(1):9, 2020.

[2] Vanessa Buhrmester, David Münch, and Michael Arens. Analysis of explainers of black box deep neural networks for computer vision: A survey. *arXiv preprint arXiv:1911.12116*, 2019.

[3] A Castellanos, Juan Cigarrán, and Ana García-Serrano. Formal concept analysis for topic detection: a clustering quality experimental analysis. *Information Systems*, 66:24–42, 2017.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[5] Chris Deotte. How to choose cnn architecture mnist. `https://www.kaggle.com/cdeotte/how-to-choose-cnn-architecture-mnist`, 2018.

[6] Anna Formica. Concept similarity in formal concept analysis: An information content approach. *Knowledge-based systems*, 21(1):80–87, 2008.

[7] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[8] Anamika Gupta, Naveen Kumar, and Vasudha Bhatnagar. Incremental classification rules based on association rules using formal concept analysis. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 11–20. Springer, 2005.

[9] Alan Julian Izenman. Linear discriminant analysis. In *Modern multivariate statistical techniques*, pages 237–280. Springer, 2013.

[10] Ashwani Jindal. Mnist classification using random forest. `https://www.kaggle.com/ashwani07/mnist-classification-using-random-forest`, 2018.

[11] Sasan Karamizadeh, Shahidan M Abdullah, Azizah A Manaf, Mazdak Zamani, and Alireza Hooman. An overview of principal component analysis. *Journal of Signal and Information Processing*, 4(3B):173, 2013.

[12] Vrushali Y Kulkarni and Pradeep K Sinha. Pruning of random forest classifiers: A survey and future directions. In *2012 International Conference on Data Science & Engineering (ICDSE)*, pages 64–68. IEEE, 2012.

[13] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[14] Andrew Maas, Quoc V Le, Tyler M O'neil, Oriol Vinyals, Patrick Nguyen, and Andrew Y Ng. Recurrent neural networks for noise reduction in robust asr. 2012.

[15] Keming Mao, Renjie Tang, Xinqi Wang, Weiyi Zhang, and Haoxiang Wu. Feature representation using deep autoencoder for lung nodule image classification. *Complexity*, 2018, 2018.

[16] Mehedi Masud, Amr E Eldin Rashed, and M Shamim Hossain. Convolutional neural network-based models for diagnosis of breast cancer. *Neural Computing and Applications*, pages 1–12, 2020.

[17] Stefan Petscharnig, Mathias Lux, and Savvas Chatzichristofis. Dimensionality reduction for image features using deep learning and autoencoders. In *Proceedings of the 15th international workshop on content-based multimedia indexing*, pages 1–6, 2017.

[18] Hussam Qassim, Abhishek Verma, and David Feinzimer. Compressed residual-vgg16 cnn model for big data places image recognition. In *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 169–175. IEEE, 2018.

[19] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

[20] BK Tripathy, DP Acharjya, and V Cynthya. A framework for intelligent medical diagnosis using rough set with formal concept analysis. *arXiv preprint arXiv:1301.6011*, 2013.

[21] Rudolf Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. In *International Conference on Formal Concept Analysis*, pages 314–339. Springer, 2009.

[22] Rudolf Wille and Bernhard Ganter. Formal concept analysis, 1996.

[23] Karl Erich Wolff. A first course in formal concept analysis. *SoftStat*, 93:429–438, 1993.

[24] Baoyao Zhou, Siu Cheung Hui, and Kuiyu Chang. A formal concept analysis approach for web usage mining. In *International Conference on Intelligent Information Processing*, pages 437–441. Springer, 2004.