Image Similarity Using Logistic Regression

Meesala. Sai Kumar
Computer Science & Engineering
Guru Nanak Institution Technical Campus
Telangana, India
meesalasaikumar1724@gmail.com

Mohammad Nayeem
Computer Science & Engineering
Guru Nanak Institution Technical Campus
Telangana, India
Mohammadnayeem1264@gmail.com

J. Yamuna
Computer Science & Engineering
Guru Nanak Institution Technical Campus
Telangana, India
jimidiyamuna@gmail.com

Mrs. B. Surekha
(Assistant Professor)
Computer Science & Engineering
Guru Nanak Institution Technical Campus
Telangana, India
Surekhait21@gmail.com

ABSTRACT –
Many machine learning algorithms, such as kernel machines, nearest neighbors, clustering, and anomaly detection, rely on distances or similarities to identify patterns in data. Before using these similarities to train a model, it is crucial to ensure they reflect meaningful relationships within the data. In this paper, we propose enhancing the interpretability of these similarities by augmenting them with explanations. To achieve this, we introduce Logistic Regression & Haar Cascade, a scalable and theoretically sound method designed to systematically decompose the output of pre-trained deep similarity models for pairs of input features. Our approach can be viewed as a composition of regression- based explanations, which previous research has shown to effectively scale to highly nonlinear models. Through extensive experiments, we demonstrate that Logistic Regression consistently provides robust explanations for complex similarity models. We also apply our method to a challenge in digital humanities: evaluating the similarity between historical documents, such as astronomical tables. In this context, Logistic Regression & Haar Cascade offers valuable insights and enhances the interpretability of a specialized, highly engineered similarity model.

Key words
Logistic Regression, Haar Cascade, Image Similarity, Machine Learning, Explainable AI, Deep Learning, Object Detection, Convolutional Neural Networks (CNN), Feature Extraction, Data Preprocessing.

1. INTRODUCTION

Building effective similarity models that integrate prior knowledge about the data and the task is a critical aspect of machine learning and information retrieval. High- quality similarity models are essential for identifying relevant items in databases and serve as the foundation for numerous machine learning approaches, including discriminative learning, unsupervised learning, and data embedding or visualization.

A key practical challenge lies in selecting the appropriate similarity model. Constructing a labeled dataset for validation can be labor-intensive, requiring meticulous examination of multiple data point pairs and assigning precise similarity scores. Alternatively, similarity models can be selected based on their performance in proxy tasks, though this approach risks issues such as overfitting to specific examples or lacking representativeness in training data, akin to the "Clever Hans" effect.

This paper proposes a more direct approach to evaluating similarity models by leveraging methods from explainable machine learning (ML). Explainable ML focuses on making model predictions interpretable for humans, with a variety of techniques developed for ML classifiers. For instance, logistic regression coefficients can reveal the relative importance of input features in making predictions, where larger coefficients signify a greater influence on the outcome. This emphasis on interpretability ensures similarity models are not only effective but also transparent and robust.

2. Related Work Area

M. Sundararajan, A. Taly, and Q. Yan (2017) introduced Integrated Gradients, an attribution method for deep networks that satisfies two key axioms: Sensitivity and Implementation Invariance, addressing weaknesses in previous approaches. Their method requires no network modifications and relies on standard gradient computations, making it simple and efficient. Demonstrated on image, text, and chemistry models, it effectively debugs networks, extracts rules, and enhances user engagement with models.

M. Tsang, D. Cheng, and Y. Li (2018) proposed a framework for detecting statistical interactions in feedforward neural networks by analyzing their learned weights. The method leverages the non- additive effects of nonlinear activation functions and encodes interacting paths in weight matrices, avoiding the need to explore an exponential solution space. It delivers state-of-the-art accuracy and efficiency, as demonstrated on synthetic and real-world datasets, highlighting the importance of discovered interactions in machine learning.

These studies by J. Kauffmann and collaborators (2019, 2020) enhance Explainable AI (XAI) by introducing methods to explain unsupervised models like clustering and anomaly detection. They propose "neuralization," where clustering models and one-class SVMs are reformulated as neural networks, enabling the use of Deep Taylor Decomposition (DTD) for efficient feature-based explanations. Their frameworks, OC-DTD and cluster explanation methods, provide insights into why data points are assigned to clusters or flagged as anomalies, outperforming traditional explanation

approaches and enriching the interpretability of machine learning models.

3. METHODOLOGY

Image similarity using logistic regression

Logistic regression, a supervised learning algorithm for binary classification, and Haar cascade, an object detection algorithm designed for detecting objects in images or videos, can be combined to enhance object detection tasks. This approach involves two stages: image classification using logistic regression and targeted object detection using Haar cascade.

First, a logistic regression model is trained on a labeled dataset of images to learn the relationship between the input features and their respective labels. Once trained, this model predicts the label of new images, categorizing them into predefined classes. The predicted labels act as a guide for the next step, where the Haar cascade algorithm is applied. Haar cascade, which uses features and cascaded classifiers to detect objects, can then focus on identifying specific objects—such as a person—within the context of the predicted label.

This combination of methods offers several advantages. By narrowing the search space for Haar cascade based on the logistic regression predictions, the object detection process becomes more efficient and accurate. For example, instead of applying Haar cascade across all possible image regions indiscriminately, it can be used selectively on images or areas likely to contain the target object, as indicated by the logistic regression output.

The synergy between these algorithms is particularly beneficial for tasks involving both image classification and object detection. Logistic regression efficiently classifies the overall context of an image, while Haar cascade specializes in precise localization of objects within that context. Beyond improved performance, this approach enhances transparency and interpretability in machine learning workflows by aligning object detection with classification outcomes.

Moreover, this method underscores a broader contribution to the design and validation of similarity-based machine learning models. By systematically integrating classification and detection techniques, it provides an efficient, informed, and human-interpretable framework for tackling complex image analysis problems.

Modules used in this project:

The project is organized into several key modules, each with a specific function. The Dataset module manages the labeled data required for training and evaluation. Importing the necessary libraries ensures all tools and dependencies are ready for the task. Retrieving the images involves loading and preprocessing image data for further processing. Splitting the dataset divides the data into training and testing subsets to train and validate the model. Building the model focuses on constructing the machine learning model architecture. Once trained, the model's performance is assessed by applying the model and plotting accuracy and loss graphs. The Accuracy on test set module evaluates the model on unseen data to measure its generalization. Finally, Saving the Trained Model ensures the model can be reused without retraining. Together, these modules streamline the machine learning workflow, from data preparation to deployment.
We will split the dataset into training and test sets, allocating 80% for training and 20% for testing.

Convolutional Neural Networks (CNNs)

The first module of the course focuses on understanding key concepts in CNNs. The objectives include:

- Understanding the convolution operation.
- Understanding the pooling operation.
- Familiarizing with the vocabulary used in CNNs (e.g., padding, stride, filter).
- Building a CNN for multi-class image classification.

Computer Vision

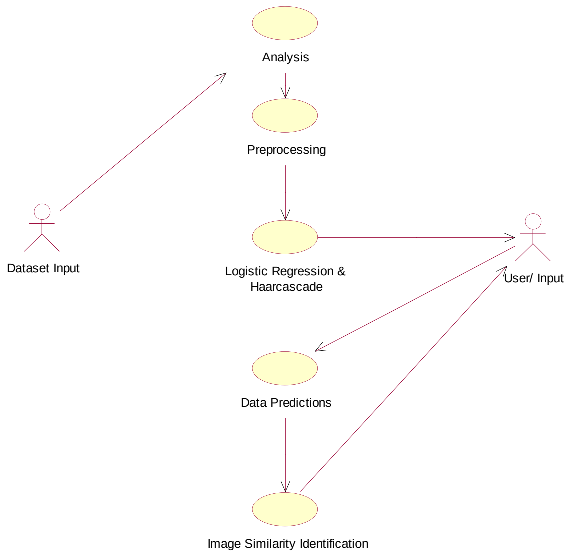
In this article, we will address several computer vision problems, such as:

- Image classification
- Object detection
- Neural style transfer

A significant challenge in computer vision is handling large input data. For example, an image of size 68x68x3 results in an input feature dimension of 12,288, and larger images (e.g., 720x720x3) would have an even higher dimensionality. Passing such large inputs through a neural network increases the number of parameters, leading to much higher computational and

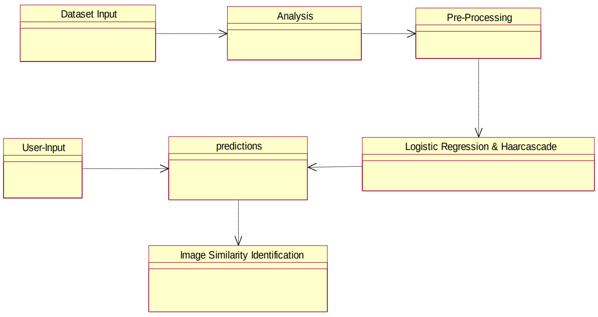
memory requirements, which can be difficult to manage.
Building the module:

To build the model, we will use the Sequential model from the Keras library. The model will leverage FaceNet, a deep convolutional neural network (CNN) designed for face recognition. FaceNet takes an input image of a face and outputs a feature vector representing the face. Trained on a large dataset of facial images, the CNN learns to extract facial features that are discriminative and invariant to variations in lighting, pose, and facial expressions. The feature vectors generated by FaceNet ensure that faces from the same person are mapped to similar vectors, while faces from different individuals are mapped to distinct vectors. This enables face recognition by comparing the distance between the feature vectors of two faces.

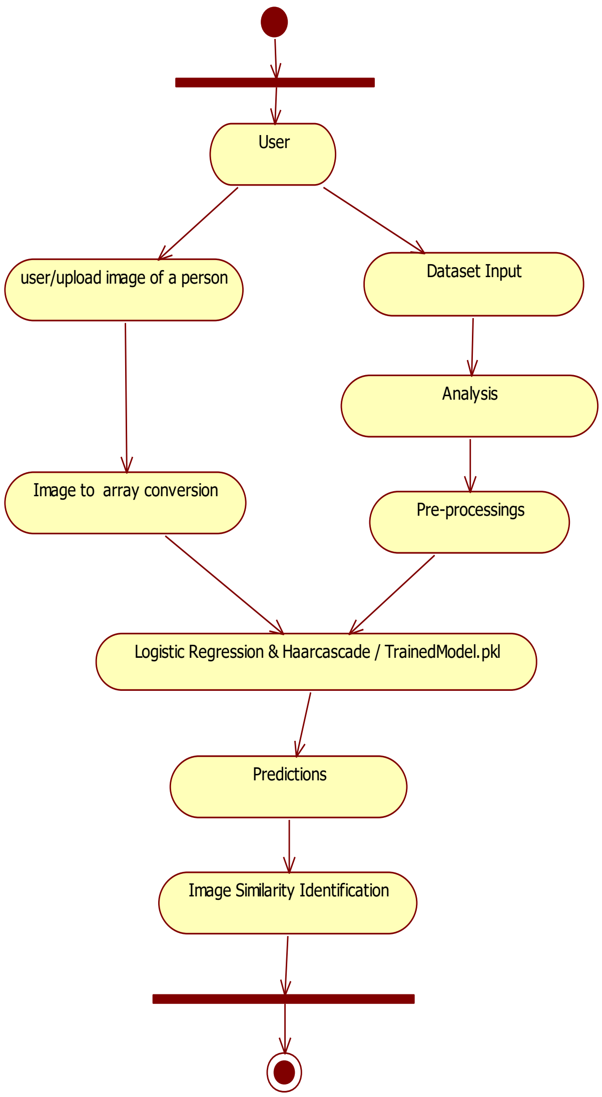

Analysis
Preprocessing
Dataset Input
Logistic Regression & Haarcascade
User/ Input
Data Predictions
Image Similarity Identification

USE CASE DIAGRAM

The main purpose of a use case diagram is to illustrate which system functions are performed by each actor. It shows the roles that actors play within the system to achieve specific objectives. In the diagram, the user is depicted as the actor, with each actor fulfilling a particular role to contribute to the overall concept.


Dataset Input
Analysis
Pre-Processing
User-Input
predictions
Logistic Regression & Haarcascade
Image Similarity Identification

OBJECT DIAGRAM


User
user/upload image of a person
Dataset Input
Analysis
Image to array conversion
Pre-processings
Logistic Regression & Haarcascade / TrainedModel.pkl
Predictions
Image Similarity Identification

ACTIVITY DIAGRAM

Activity diagrams are graphical representations that illustrate the workflow of step-by-step activities and actions within a system. They support various elements such as choices, iterations, and concurrency, allowing for a detailed depiction of how different tasks and processes flow. In the Unified Modeling Language (UML), activity diagrams are used to describe both business and operational workflows, showing how components within a system interact and progress through a sequence of actions. These diagrams highlight the overall flow of control, making it easier to visualize the interactions and decisions that occur throughout the system's processes.

4. Results

In this approach, the goal is to develop a distance metric that measures the similarity between two images, x1 and x2. The key innovation is the use of CNN attention to generate similarity model explanations, which can also be leveraged to enforce trainable constraints during model training. This technique not only enhances the model's ability to explain its predictions but also improves its generalizability. By incorporating similarity attention, the model can generate attention maps, which visually highlight the regions of the input images that influence the similarity decision. These maps offer valuable insights into why the model predicts that the two images satisfy the similarity criterion, providing greater transparency and interpretability in the similarity evaluation process.

The use of CNN attention in similarity models provides significant advantages for both model interpretability and performance. By generating attention maps, we gain a deeper understanding the decision-making process, revealing which features of the images are most relevant to the model's similarity predictions. This transparency allows for better debugging and refinement of models, ensuring that they focus on the correct aspects of the input data. Furthermore, the ability to enforce trainable constraints improves the model's robustness and generalization across diverse image data, making it more adaptable to real-world scenarios. Overall, CNN attention enriches the model with both explainability and enhanced predictive power, enabling more informed and reliable similarity analysis

Logistic regression is primarily used for numerical data analysis and prediction, offering a straightforward approach to binary classification tasks. On the other hand, Haar cascade is specialized for object detection in images and videos. It works by extracting a set of features from images and using them to train a classifier that can detect the presence or absence of specific objects or features.

The simplicity and computational efficiency of logistic regression and Haar cascade make them excellent choices for scenarios with strict resource constraints or real-time processing needs. While logistic regression

is effective in tasks requiring numerical analysis, Haar cascade provides a powerful solution for object detection, particularly in applications such as face detection or vehicle recognition in videos. Although less powerful than deep learning models like CNNs, these techniques offer an accessible balance of accuracy and efficiency, making them practical tools in many real-world settings where computational resources may be limited.

5. Conclusion

We have introduced a theoretically robust method for explaining similarity in terms of pairs of input features. Our approach, based on Logistic Regression, can be viewed as a composition of regression computations, allowing it to inherit the model's robustness and broad applicability while extending its utility to the new domain of similarity explanation.

The effectiveness of Haar cascade was demonstrated in the context of similarity analysis, as implemented by Logistic Regression. Haar cascade was able to predict transfer learning capabilities and identify instances of ‘Clever Hans’ predictions, where the model may rely on spurious correlations. Additionally, Haar cascade proved valuable in a practical problem in the digital humanities, where, even with limited data, it outperformed general models by demonstrating the superiority of a task-specific similarity model. This highlights Haar cascade's ability to efficiently tackle domain-specific challenges, making it a useful tool for both similarity explanation and real-world applications with constrained datasets.

6.REFERENCES

1.A. Zien, G. R€atsch, S. Mika, B. Sch€olkopf, T. Lengauer, and K.-R. M€uller, “Engineering support vector machine kernels that recognize translation initiation sites,” Bioinformatics, vol. 16, no. 9, pp. 799–807, 2000.

2.C. D. Manning, P. Raghavan, and H. Sch€utze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge Univ. Press, 2008.

3.A. Nierman and H. V. Jagadish, “Evaluating structural similarity in XML documents,” in Proc. Int. Workshop Web Databases, 2002.

4.E. Pampalk, A. Flexer, and G. Widmer, “Improvements of audio- based music similarity and genre classification,” in Proc. 6th Int. Conf. Music Inf. Retrieval, 2005.

5.P. Willett, J. M. Barnard, and G. M. Downs, “Chemical similarity searching,” J.Chem.Inf.Comput. Sci., vol. 38, no. 6, pp. 983–996,
1998.

6.C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics). Berlin, Germany: Springer-Verlag, 2006.

7.B. Sch€olkopf and A. J. Smola, Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Cambridge, MA, USA: MIT Press, 2002.

8.J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in Proc. 5th Berkeley Symp. Math. Statist. Probability, 1967.

9.A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: A review,” ACM Comput. Surv., vol. 31, no. 3, pp. 264–323, 1999.

10.J. Shi and J. Malik, “Normalized cuts and image segmentation,” IEEE Trans. Pattern Anal. Mach.

Intell.,vol.22,no.8,pp.888– 905,Aug.2000.
11.B. Sch€olkopf, J. C. Platt, J. Shawe- Taylor, A. J. Smola, and R. C. Williamson, “Estimating the support of a high-dimensional distribution,” Neural Comput., vol. 13, no. 7, pp. 1443–1471, 2001.