World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:18, No:11, 2024

Integrating AI Visualization Tools to Enhance Student Engagement and Understanding in AI Education

Yong W. Foo, Lai M. Tang

***Abstract*—**Artificial Intelligence (AI), particularly the usage of deep neural networks for hierarchical representations from data, has found numerous complex applications across various domains, including computer vision, robotics, autonomous vehicles, and other scientific fields. However, their inherent "black box" nature can sometimes make it challenging for early researchers or school students of various levels to comprehend and trust the results they produce. Consequently, there has been a growing demand for reliable visualization tools in engineering and science education to help learners understand, trust, and explain a deep learning network. This has led to a notable emphasis on the visualization of AI in the research community in recent years. AI visualization tools are increasingly being adopted to significantly improve the comprehension of complex topics in deep learning. This paper presents an approach to empower students to actively explore the inner workings of deep neural networks by integrating the student-centered learning approach of flipped classroom models with the investigative capabilities of AI visualization tools, namely, the TensorFlow Playground, the Local Interpretable Model-agnostic Explanations (LIME), and the SHapley Additive exPlanations (SHAP), for delivering an AI education curriculum. Integrating these two factors is crucial for fostering ownership, responsibility, and critical thinking skills in the age of AI.

***Keywords*—**Deep Learning, Explainable AI, AI Visualization, Representation Learning.

I. Introduction

AI, in particular, Deep Learning (DL), has become the most widely used computational approach due to its impressive results in solving complex problems in diverse fields. DL refers specifically to the use of neural networks with multiple layers, where each layer automatically learns and extracts complex features from the data. This is also known as representation learning, whereby the primary goal is to capture essential patterns or representations within the data. Representation learning inside the neural networks is organized hierarchically, with features extracted at increased abstraction at each layer. The simpler patterns learned at lower levels lay the foundation for more abstract and complex patterns learned at the higher levels. At the networks' final layers, the data's essential characteristics are captured and formatted in a way suitable for classification, clustering, or other Machine Learning (ML) predictions.

Among the types of DL networks, convolutional neural networks (CNNs) stand out as one of the most extensively employed DL models. The CNN architecture often serves as a foundational framework for early researchers and students to understand the complexity of DL networks. The CNN model presents an excellent learning framework for various DL perspectives, such as network architecture, tensor formulation and manipulation, backpropagation computation, and hierarchical representation. As information flows through the networks, each layer automatically learns and extracts complex features from the data by optimizing the network parameters. This optimization is achieved by minimizing an objective function, utilizing backpropagation, and being guided by gradient descent algorithms. The iterative adaptive learning process gradually converges, creating representations that prove effective for the given task.

However, DL models are commonly characterized as "black box" models as the inner workings of the models lack transparency and are challenging to comprehend. This inherent lack of transparency poses a significant obstacle for a novice student to understand the intricacies of DL networks. The complex feature maps and internal representations are especially hard to visualize. Undoubtedly, this would hinder the student's ability to effectively grasp the underlying principles or trust the results they produce [1]. Overcoming the steep learning curve of AI requires a student-centered learning approach in the education curriculum, combining theoretical knowledge with practical sessions that leverage visualization and interactivity tools to offer a more immersive and experiential understanding of complex DL concepts and AI [2].

Wright argues that a student-centered learning approach that actively involves students leads to increased ownership of their learning process and enhanced engagement [3]. Sewagegn and Diale [4] put forward that when students feel empowered through autonomy and supportive feedback, they demonstrate higher levels of engagement in classroom activities and show greater academic achievement.

This paper presents a flipped classroom model integrating visualization tools and Explainable AI (XAI) techniques, specifically, the TensorFlow Playground, the LIME [5], and the SHAP [6] as core educational resources for exploring and demystifying complex DL models. TensorFlow Playground provides an interactive web-based platform designed to facilitate a hands-on understanding of DL. LIME and SHAP serve as XAI methods that reveal insights into how DL models arrive at their predictions, offering a more transparent view of these typically opaque processes. Collectively, the tools support learners in actively engaging with AI concepts and deepening their comprehension of DL's decision-making mechanics.

Y. W. Foo is with the School of Engineering, Nanyang Polytechnic, 180 Ang Mo Kio Ave 8, Singapore 569830 (corresponding author, phone: 65-6550-0962; e-mail: Foo_Yong_Wee@nyp.edu.sg).

L. M. Tang is with the School of Computing Science, University of Glasgow, 10 Dover Drive Singapore 138683 (e-mail: laimeng.tang.2@glasgow.ac.uk).

The rest of the paper is structured as follows: Section II provides a foundational understanding of the CNN model. Section III offers an overview of the selected tools and their relevance in AI education. A comparative analysis of the selected tools, highlighting their strengths, weaknesses, and unique features, is discussed in Section IV. Section V describes a proposed flip-classroom model integrating with the AI visualization tools. The latest trends and advancements in AI education are explored in Section VI. Lastly, Section VII concludes and summarizes the key insights and recommendations.

II. CNN Model Architecture

There are many types of DL networks, such as Recursive Neural Networks (RNNs), Gated Recurrent Units (GRUs), Long Short-Term Memory (LSTM) networks, Deep Neural Networks (DNNs), CNNs, auto-encoders, and Generative Adversarial Networks (GANs). This paper focuses on the CNN, which is the most widely employed model [7] in the field of computer vision.

Several CNN models have been developed for various computer vision tasks. The commonly known ones are AlexNet, Residual Network (ResNet), GoogLeNet (Inception), and DenseNet, to name a few. The CNN architecture has multiple layers, each consisting of interconnected nodes. The nodes receive input data and perform computations using activation functions before passing the results to the next layer [8]. As data flow from the input to the output layer, the connection weights and biases (also known as parameters) associated with the node connections are learned. The activation functions, such as the ReLU (Rectified Linear Unit), sigmoid, and tanh, introduce non-linearity to the network, allowing it to learn complex patterns. The parameters are adjusted using the backpropagation algorithm, which consists of forward and backward passes during training. Gradients of the loss with respect to the weights are computed using a loss function, which measures the difference between the predictions at the output and the actual target values. This information is used to update the parameters in the backward direction, guided by optimization algorithms such as gradient descent. This process is repeated until this loss is minimized or a state of convergence is achieved where further parameter adjustments do not contribute to the model's performance.

Fig. 1 shows a CNN architecture, distinguished by its highly efficient and organized structure, characterized by several layers: the input layer, the convolutional layers, the pooling layers, the flattening layers, the fully connected layers, and the output layer; and key components: activation functions, loss functions, backpropagation, optimization algorithms, and regularization.

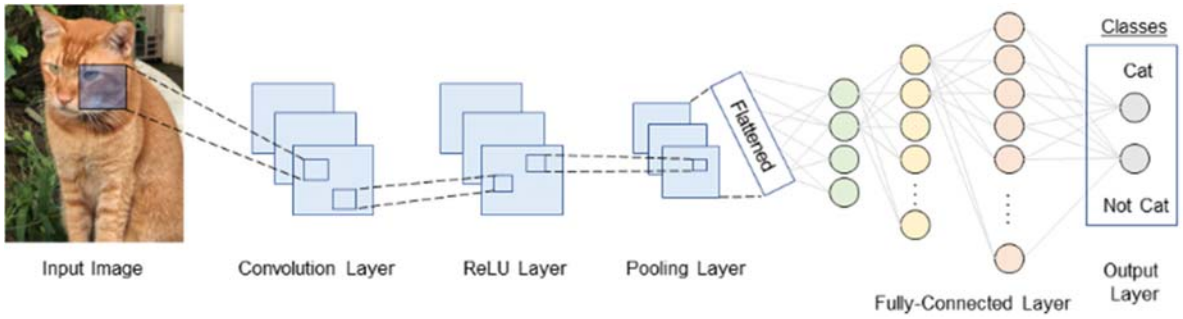


Fig. 1 The CNN Architecture

A. Input Layer

The input layer of a CNN typically comprises an image or raw pixel values, expressed as a three-dimensional matrix m x m x r, where m denotes the height and the width (equal) and r denotes the depth, also known as the channel number. For a color (RGB) image, r is equal to three.

B. Convolutional Layers

The convolutional layers are responsible for extracting hierarchical representations of the input. Each layer consists of several kernels (filters) denoted by k. Like the input image, filters have three dimensions, n x n x q, where $n < m$ and $q \leq r$. Specifically, filters contain the weights, W^k and biases, b^k (parameters) of the local connections. The values (weights) of the filters are randomly initialized and adjusted at each training epoch, attempting to detect patterns and features from the input image. Feature extraction is achieved by sliding the filters over the input data (image) and convolving to produce k feature maps, h^k with a size of $(m- n+1)$ at the convolutional-layer output. The feature maps, h^k, are calculated based on a dot product between its input (image) and the weights (filter), where the results are then subjected to the activation function, f, expressed in (1) [7]:

$$h^k = f(W^k * x + b^k) \quad (1)$$

where W^k and b^k are the weights and biases, respectively, and f is the activation function.

C. ReLU Layer - Activation Function

The purpose of activation functions is to introduce non-linearity into the model. These functions are applied in the convolutional and fully connected layers, allowing the model to learn more complex features. Examples of activation functions are the Sigmoid, Tanh, and ReLU functions. The ReLU function represented in (2) is most frequently used in CNN. It takes the values of x and converts them to positive numbers.

$$f(x) = \max(0, x) \quad (2)$$

D. Pooling Layers

This layer is responsible for down-sampling every feature map. The objective of the pooling layer is to accelerate the training process and avoid overfitting by reducing the parameters. There are several pooling techniques, such as average, min, or max pooling. These techniques create smaller feature maps while maintaining most of the dominant information.

E. Flattening Layer

The flattening layer is the layer before the fully connected layers. It is tasked with transforming the preceding layers into a one-dimensional vector, preparing it for fully connected layers.

F. Fully Connected Layers

After flattening, the fully connected layers take the input (vector) from the feature maps created in the last convolutional or pooling layer and pass it to fully connected conventional neural network layers to make predictions.

G. Output Layer

This layer consists of nodes corresponding to the number of classes where the final predictions are made based on the learned features from the previous layers.

H. Loss Functions

Loss functions are applied at the output to compute the predicted error generated across the training samples in the CNN model. The predicted error is the difference between the predicted output and target values.

I. Backpropagation, Optimization Algorithms, Learning Rate and Regularization

Backpropagation is a learning algorithm commonly used in neural network training to improve the network's accuracy by minimizing the error between the predicted and actual outputs. Backpropagation calculates the gradient of the loss function with respect to the network parameters. Based on the obtained error, optimization algorithms such as Gradient Descent and Adam are used to adjust the parameters to reduce the error in the reverse direction of the gradient (backward pass). The learning rate is a tunable hyperparameter that determines the magnitude of adjustments made. Regularization techniques, such as Dropout and L1 and L2 Regularization, are often employed to prevent overfitting and improve the network's generalization ability.

III. Overview of Selected Tools in AI Education

DL can be challenging for students grappling with the depth of concepts such as CNN architecture, activation functions, and optimization. In addition, the "black box" CNN, with its complex connectivity of neurons and calculations, can pose significant learning challenges for novice learners. Specifically, the complex feature maps and internal representations are hard to visualize, hindering the student's grasp of the underlying principles. XAI techniques as ML visualization tools can address this challenge by providing a detailed exploration of the design to shed light on the intricate workings of CNNs. In this section, we present an overview of the XAI techniques, in particular, the TensorFlow Playground, the LIME, and the SHAP, for uncovering the "black box" decision-making processes of CNNs in ImageNet classification.

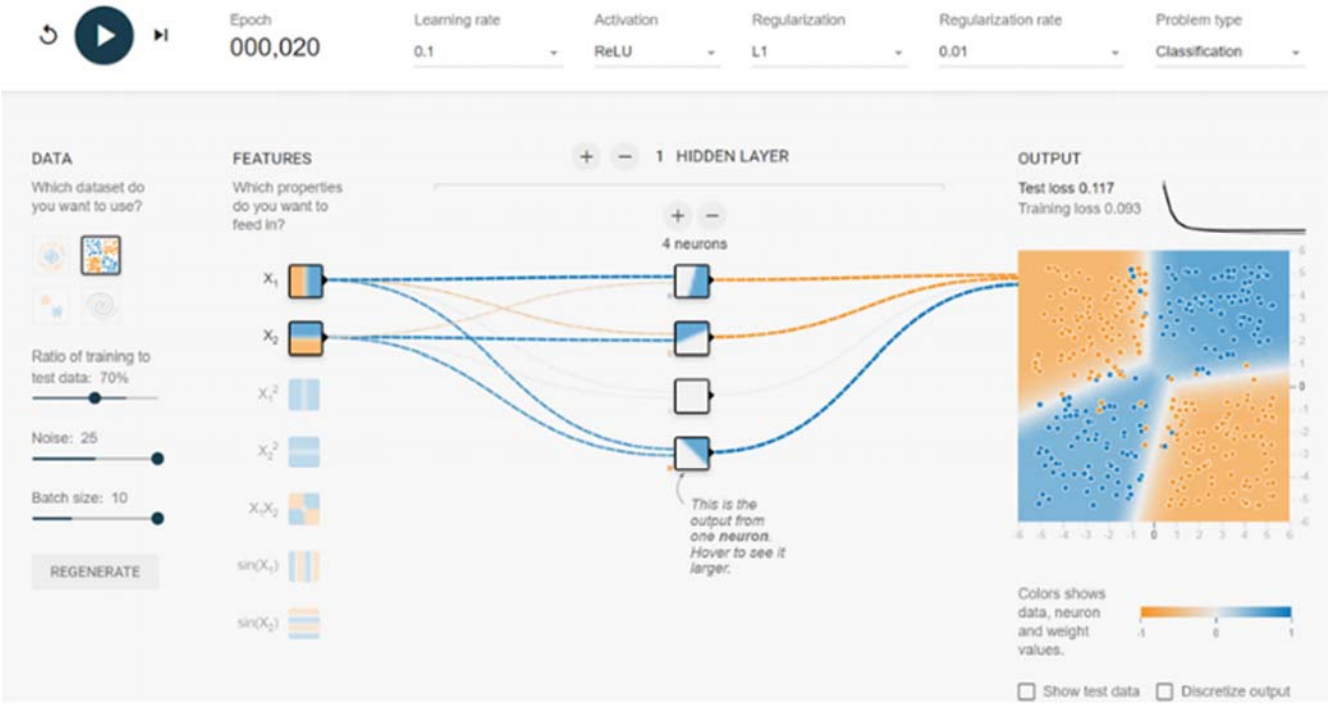
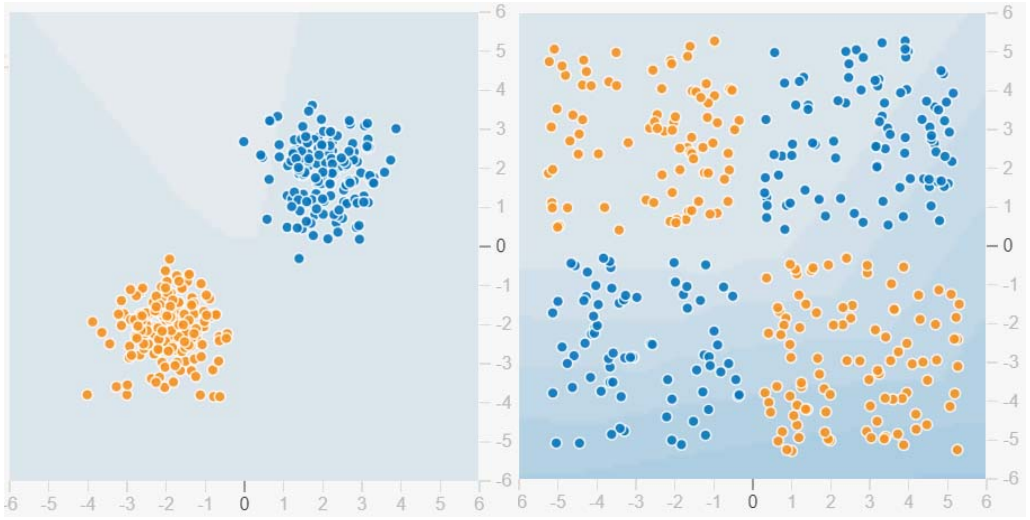


Fig. 2 The TensorFlow Playground Web Interface

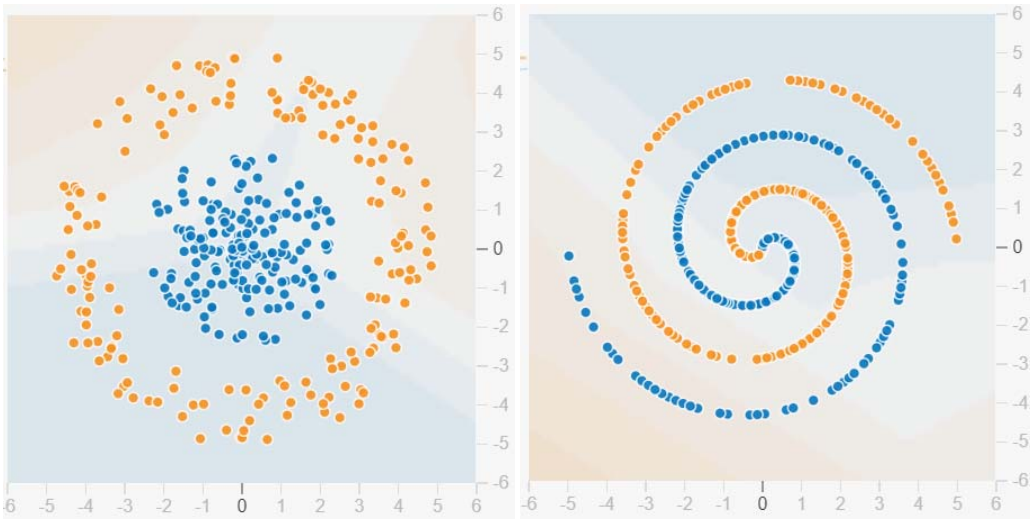
A. TensorFlow Playground

The TensorFlow Playground [8] is an open-source web-based interactive platform developed by Google that allows users to experiment visually with neural networks. As a tool for introducing the fundamental concepts of neural networks, its graphical interface allows for an interactive learning experience. It uses visual representation to show how neurons are connected, how weights, biases, and activation functions interact, and how information flows through the network. The tool provides an intuitive way to adjust parameters such as the number of hidden layers, the number of neurons in each layer, and other hyperparameter settings to see how they affect the training of a neural network. Students can receive real-time visual feedback on how the model identifies patterns from data, which makes learning more engaging and effective. Figs. 2 and 3 illustrate the TensorFlow Playground's high degree of interactivity and configurability. Fig. 2 shows the visually intuitive interface of the tool.

Two problem types are presented: regression and classification. Figs. 3 (a)-(d) show four datasets for the classification problems at different levels of complexity. In general, orange shows negative values while blue shows positive values. The distinct colored dots in the dataset represent a binary classification problem, such as Class A and B. The objective of the tools is to generate a neural network with appropriate architecture and hyperparameters to separate the dots belonging to Class A from those belonging to Class B.



(a) Dataset 1 (b) Dataset 2



(c) Dataset 3 (d) Dataset 4

Fig. 3 Datasets with Binary Classification Problems

The dataset can be refreshed at the click of the 'Generate' button and allows for incorporating noise levels up to 50%, splitting into various ratios of training and testing subsets, and configuring different batch sizes. All these adjustments can be made using the intuitive slider interface, located under the 'DATA' column, as depicted on the left of Fig. 2.

The visual representation located under the 'FEATURES' and 'HIDDEN LAYER' columns in Fig. 2 illustrates a network architecture that can be used to demonstrate how neurons are connected and how information flows through the network. The network components, such as the input layers, hidden layers, and output layers and their respective neurons, can be modified through the respective + and – icons near the top of the website. The impact of these changes on model behavior can be instantly observed in the network's capacity to capture intricate patterns within the input data. As alterations are made using the provided icons, the subsequent modifications in the network's structure dynamically shape its ability to learn and generalize, providing an interactive and insightful platform for exploring the nuances of neural network design. The model's ability to capture non-linear patterns can be demonstrated by selecting various activation functions and observing how they influence the network's performance.

The importance of hyperparameter tuning, such as the learning rate, batch size, regularization, and regularization rate, can be experimented with different values to observe how these changes affect the model's convergence and accuracy. Using the tool's visual representations, network training can be paused to explain backpropagation, how weights are updated during optimization, and how the training and testing loss decreases over time as the model learns. The weights update can be explained through the colored connections, where blue shows that the network is assigning a positive weight and orange a negative weight, with the thickness of the connection representing more positive or negative values. How well the network predicts can be visualized at the output through the background color covering the orange and blue dots. The color's intensity shows how confident that prediction is.

The concept of overfitting and how it can be mitigated can be investigated by applying L1 or L2 regularization techniques. Learners can also experiment with dropouts by removing neurons or hidden layers and then observe the impact of regularization on the model's generalization ability.

In summary, the TensorFlow Playground can be effectively used for AI education in introducing the basic concepts of neural networks.

B. Local Interpretable Model-Agnostic Explanations

LIME addresses this interpretability gap by offering local explanations providing insights into why a specific image was classified in a particular way. It employs a model-agnostic strategy, creating locally accurate interpretations by training a straightforward interpretable model, such as linear regression, on perturbed samples centered around the specific instance of interest. This involves introducing variations to the input data and examining the resulting alterations in the model's predictions [5]. In the image classification task, LIME primarily provides local interpretability by generating explanations for individual predictions on local super pixel regions of specific images. It perturbs the input image to create a local surrogate model that approximates the behavior of the black-box model for that particular instance [9].

LIME's methodology for image classification involves several processes. First, it chooses an image for interpretation. This instance will serve as the focal point for generating local explanations. Next, it introduces small and controlled perturbations to the selected image to create a dataset of slightly modified versions. Such image perturbations could include changes in pixel values, rotations, or other transformations. LIME then feeds the perturbed images, along with the original, through CNN to obtain predictions. The model's outputs for each perturbed image are then recorded. The subsequent step is to fit an interpretable, often linear, model to the perturbed instances and their corresponding model predictions. This surrogate model approximates the CNN's behavior within the chosen image's local neighborhood. The final step is to generate an explanation by analyzing the coefficients of the surrogate model to understand the influence of different features (pixels) on the model's prediction for the original image. The advantages of LIME for Image Classification lie in such local interpretability capability. This makes LIME excel in providing detailed explanations for individual image predictions. This local interpretability is valuable for understanding model decisions on a case-by-case basis.

LIME can be applied to any black-box model as a model-agnostic approach, making it versatile for interpreting a wide range of CNN architectures. The interpretable surrogate model generated by LIME, often a linear regression model, offers user-friendly explanations that are easy to understand, even for non-experts. By revealing the contribution of different pixels to the model's decision, LIME aids in making the decision-making process of CNNs more transparent and interpretable.

To illustrate how LIME works using an example, we consider Figs. 4 (b)-(d), which show the results of applying LIME processes to the original image shown in Fig. 4 (a). This color image consists of two main objects: a cat and a mouse.

Here, we apply LIME on a standard InceptionV3 pre-trained model for illustration purposes. Fig. 4 (b) shows the generated explanation as super pixels of the predicted class having the most positive weight values with the rest of the pixels excluded. In contrast, Fig. 4 (c) shows the predicted super pixels together with the rest of the original pixels. The explanation weights are shown as a heatmap for visualization purposes in Fig. 4 (d). These figures show that from a model's output prediction, we can fit them into an interpretable model. This surrogate model approximates the CNN behavior within the local neighborhood of the chosen image. To generate an explanation of the model, we need to analyze the coefficients of the surrogate model to understand the influence of different features (pixels) on the model's prediction for the original image. In this example, the local interpretability of the cat's head and surrounding fur explains the successful classification prediction of the model.

C. Shapley Additive Explanations

SHAP employs principles from cooperative game theory principles to equitably assign the contribution of each feature to the model's output. It calculates the average contribution of each feature considering all conceivable combinations [6]. SHAP offers a dual capability in image classification by providing local and global interpretability [9]. The local

interpretability explanations can be obtained for individual image predictions, while global interpretability provides insights into feature importance across the entire dataset. The Shapley values, which represent the average contribution of each feature (pixel) to the model's output, allow for a more holistic understanding of the importance of different image regions. These values can be computed for individual predictions, offering insights into local interpretability or averaged across the complete dataset, thereby providing a broader perspective on global interpretability [10].

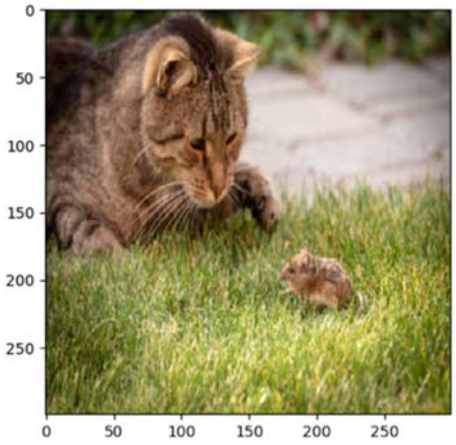


Fig. 4 (a) Input image

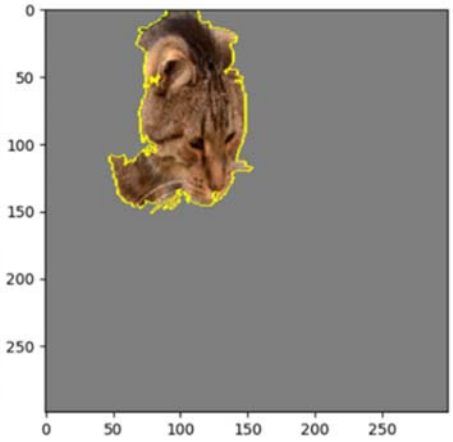


Fig. 4 (b) Super pixel outline

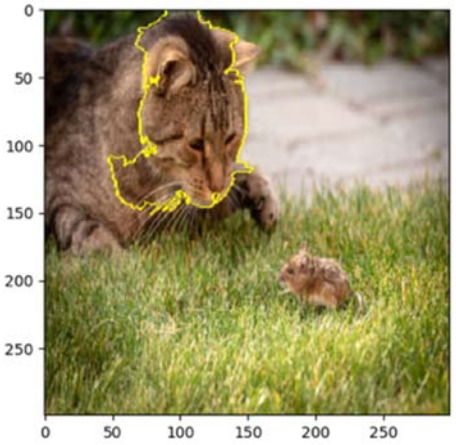


Fig. 4 (c) Image with Super pixel

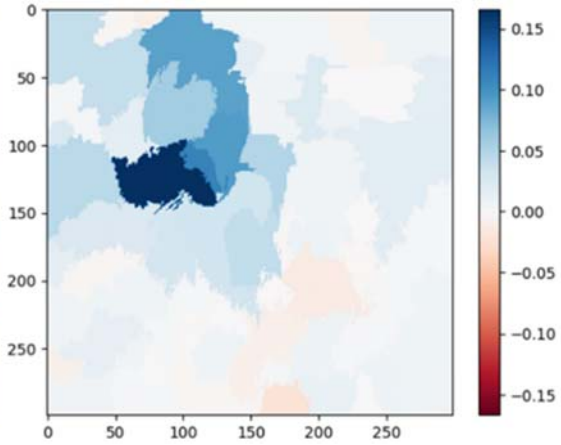


Fig. 4 (d) Heatmap

SHAP works by first establishing a background dataset representing a range of images similar to the dataset used to train the CNN. This dataset serves as a reference to compute feature importance. This technique then calculates Shapley values, which represent the average contribution of each pixel to the model's output, using the background dataset and the CNN model. This involves evaluating the model's prediction for all possible combinations of features. Then, it applies SHAP values to explain the prediction of a specific image. The positive or negative SHAP values assigned to each pixel indicate its influence on the model's decision process. In the final step, we can visualize the SHAP values by highlighting the regions of the image that significantly contribute to the model's output. This can be done using heatmaps or other visualization techniques.

To illustrate how SHAP works, we consider Figs. 5 (a) and (b), which show the results of applying SHAP processes to the original image consisting of one main object, namely a great grey owl. Here, SHAP is applied to a standard RESNET50 pre-trained model for illustration purposes. The four generated explanations corresponding to the top four predicted classes with the most positive values are shown as super pixels, with the rest of the pixels blurred in Fig. 5 (a). The explanations with finer details generated are shown in Fig. 5 (b). These figures show that from a model's output prediction, we can use the SHAP values to explain the prediction of a specific image. The positive or negative SHAP values assigned to each pixel indicate its influence on the model's decision. In this image example, the top four predicted classes are Great Grey owl, Peacock, Ostrich, and Prairie Chicken, respectively. The most important feature for the correct classification results is the "big, rounded eyes and eyelids" over the bird's head, highlighted as red super pixels. These features learned provide a crucial explanation for the correct classification result of the CNN network model.

IV. Comparative Analysis and Recommendations

LIME emerges as a valuable tool for shedding light on the black-box nature of CNNs in image classification tasks [9]. Its ability to provide local, interpretable explanations makes it a practical choice for understanding model predictions on individual images. However, LIME can be computationally expensive, especially when dealing with high-dimensional data like images [5]. It may require generating a substantial number of perturbed samples for accurate interpretation [9].

Fig. 5 (a) Four classes and their respective generated explanations with the most positive values shown as super pixels

Fig. 5 (b) The red super pixels highlighting the most important feature for the correct classification results: “big, rounded eyes and eyelids” over the bird’s head

Regarding parameter tuning, the choice of parameters, such as the number of perturbed samples and the complexity of the surrogate model, can impact the quality of LIME explanations. Hence, careful tuning is essential for optimal results. In the final analysis, LIME assumes that the decision boundary is locally linear, which might not always hold true. Students should be aware of the potential limitations of the local approximation provided by the surrogate model.

Similar to LIME, SHAP values can be computationally expensive, especially for complex models and large datasets [11]. However, approximation algorithms such as FAST SHAP [11] and TreeSHAP for tree-based models can be employed to mitigate computational costs. In addition, interpreting and visualizing SHAP values for images can be challenging due to the high dimensionality of pixel data. Hence, effective visualization techniques are crucial for making the results accessible and understandable. In terms of its dependency on model properties, while SHAP is model-agnostic, its efficiency and accuracy can vary based on the model’s properties. It performs well with models that exhibit certain characteristics, such as additivity [4].

In applying these techniques during class, one should be mindful of the computational cost and the assumptions inherent in LIME’s methodology and SHAP. While LIME offers a promising avenue for bridging the gap between complex CNNs and human interpretability in image classification scenarios, SHAP provides both local and global interpretability, along with fair feature attribution, which positions SHAP as a valuable asset in the quest for transparent and interpretable ML models. In an ever-evolving deep learning domain, SHAP offers a principled approach to understanding the intricate decisions made by CNNs in the realm of image classification.

V. Flip-Classroom Model with AI Visualization Tools for Deep Learning

This paper proposed a flipped classroom model for delivering an AI education curriculum due to its well-supported student-centered learning approach. Structurally, the flipped classroom model fosters active learning activities, which increase ownership and responsibility for learning, leading to greater student engagement [12]. Naik [13] highlighted the model’s ability to promote learner autonomy, critical thinking, and self-directed learning through active engagement and collaborative activities. Estrada et al. [14] argue that providing students with a sense of control and autonomy can increase student engagement, interest, and, ultimately, better learning

outcomes. Zainuddin and Perera's study [15] reveals that the model enhances student motivation and learning gains by fostering ownership and responsibility for the learning process through active engagement and personalized learning opportunities.

In the proposed flipped classroom model, the investigative nature of TensorFlow Playground, LIME, and SHAP is leveraged to empower students to delve into the inner workings of AI models. The tools' powerful interactive and visualization features allow for exploring the internal patterns of the black-box models. These visualization tools can explain the interpretation of the model's decision-making process and how it derives the filters activated by images in various layers. The increased interpretability of the model can significantly improve learning and trust in the model and its application in the real world.

A. Module Description and Learning Outcomes

In the proposed 'Flip-Classroom Model with AI Visualization Tools for Deep Learning,' students delve into the foundational theory of data-driven learning and explore various state-of-the-art deep learning models. The module extensively covers cutting-edge methods, including CNNs, recurrent neural networks, transformers, GANs, and reinforcement learning. Emphasis is placed on a flipped classroom approach, empowering students to actively engage with the content through pre-class materials. The AI visualization and XAI tools are introduced to enhance comprehension. Students participate in collaborative activities, discussions, and hands-on exercises during classroom sessions, fostering a deeper understanding of the deep learning concepts. The module concludes with implementing deep learning models, enabling practical application in real-world scenarios.

B. Module Learning Outcomes

The module learning outcomes include analyzing the inner workings of deep learning models to understand model behavior, interpreting how DNNs make predictions through XAI tools, applying XAI techniques to analyze CNN, RNN, and GAN architectures, critically evaluating different XAI techniques to identify the most appropriate method for explaining and interpreting real-world implementation of DL models.

C. Pre-class Activities (Online Resources)

The pre-class activities include asynchronous online video lectures introducing the basic concepts of CNNs, RNNs, and GANs. Interactive tutorials are proposed to guide students through basic DNN training using TensorFlow Playground. Additional reading materials, such as articles or curated blog posts introducing XAI, LIME, and SHAP, are also provided.

D. In-class Activities (Hands-on Session)

For in-class activities, a warm-up exercise is designed for students to revisit the TensorFlow Playground trained model they explored at home. Subsequently, students are divided into groups for the first activity. In this activity, groups are assigned a pre-trained CNN model. Using LIME, students generate visual explanations for the model's predictions on specific images. The instructor explains how these tools enable exploring and discovering hidden patterns within DNNs. Students are given time to discuss and compare the interpretations within their groups while also engaging with the materials at their own pace, pausing or re-reading as needed. The instructor reinforces learning by clarifying concepts, addressing misunderstandings, and facilitating deeper material exploration.

Another in-class activity is designed to introduce the GAN. Students explore GAN utilizing SHAP to visualize the internal representations used by the GANs to generate outputs. In their respective groups, students discuss the challenges of explaining how GANs make decisions or why they generate specific outputs. After gaining a foundational understanding, learners can then brainstorm potential applications of GANs and explore innovative ways to implement them. By allowing learners greater autonomy over their learning pace and engaging them in activities that promote higher-order thinking skills, instructors create a more learner-centered environment. Personalized guidance can be offered to those needing extra support, helping to ensure all learners have a strong grasp of complex concepts and develop the competency to apply these skills within diverse and complex applications.

E. Class Discussion

A group discussion is proposed focusing on comparing and contrasting the effectiveness of LIME and SHAP in their application to specific tasks. The discussion could also focus on analyzing the limitations of XAI tools and the importance of critical interpretation. To develop critical thinking skills, a student debate is proposed on the ethical implications of XAI in applications like facial recognition.

F. Evaluation

A pre-class quiz is designed to assess understanding of core concepts introduced in video lectures and resources. At the end of the class, students are required to individually submit a reflection report summarizing their XAI exploration findings and key learning points on the class discussion.

The proposed flipped classroom model integrating XAI tools can be easily adapted to other DL topics. The key is choosing the XAI tools appropriate for the specific DNN architecture and available resources.

VI. Future Direction

This research will continue with plans to conduct a study to examine the effects of the flipped classroom strategy. The students will be divided into two groups: an experimental flipped classroom group and a control group. The experimental group shall be taught the concept of DL via a flipped classroom, whereas the control group taught via the traditional lecture-tutorial-lab strategy. Both quantitative and qualitative approaches will be utilized in the experiments.

As the landscape of AI education continues to evolve, there is a need to delve into the exploration and synthetization of advanced tools, technologies, methodologies, systems and

platforms, and educational processes to propel AI learning experiences to the next level. Moving forward, interactive learning platforms can be combined with XAI visualization tools. This includes the integration of interactive learning platforms into the XAI tools to simulate real-world scenarios, providing students with hands-on experience in applying AI in computer vision applications. Tools such as Augmented Reality (AR), Virtual Reality (VR), and Edge Computing can be incorporated to explore their potential in creating immersive learning environments, allowing students to visualize complex concepts in AI and computer vision and enabling students to work with resource-efficient models and fostering a deeper understanding of deployment considerations.

In addition, adaptive learning systems can also be implemented to tailor educational content based on individual student progress, ensuring a personalized and efficient learning experience. Collaborative learning platforms can be introduced to facilitate knowledge sharing and teamwork, mirroring real-world scenarios in developing AI and computer vision solutions.

Finally, further research will be embarked to explore other pedagogical approaches in AI education to cater to diverse learning styles, ensuring inclusivity in AI education.

VII. Conclusion

This paper presented an approach in AI education by integrating AI visualization tools, specifically TensorFlow Playground, LIME, and SHAP, as integral components of a flipped classroom model for a DL curriculum. The unique capabilities of TensorFlow Playground provide an interactive platform for students to experiment with DL concepts, while LIME and SHAP, as XAI techniques, uncover the black-box nature of neural networks. This integration into a flipped classroom model facilitates active student engagement, enabling exploration and visualization of internal representations critical to understanding how black-box DL models such as CNNs process and extract features from images. An outline of a flipped classroom model was proposed, including the module description, learning outcomes, and pre- and in-class activities, providing a student-centered visual and interactive learning experience fostering ownership, responsibility, and critical thinking skills preparing students to navigate the complexities of modern AI landscapes.

References

[1] Chatzimparmpas, A. and Martins, R. M. and Jusufi, I. and Kucher, K. and Rossi, F. and Kerren, A., "The State of the Art in Enhancing Trust in Machine Learning Models with the Use of Visualizations," vol. 39(3), pp. 713–756, Computer Graphics Forum, Wiley, Jun. 2020, ISSN 1467-8659, http://dx.doi.org/10.1111/cgf.14034, DOI 10.1111/cgf.14034.

[2] Gresse von Wangenheim, C., Hauck, J.C.R., Pacheco, F.S. et al., "Visual tools for teaching machine learning in K-12: A ten-year systematic mapping." Educ Inf Technol vol. 26, pp. 5733–5778 (2021). https://doi.org/10.1007/s10639-021-10570-8.

[3] G. B. Wright, "Student-Centered Learning in Higher Education," International Journal of Teaching and Learning in Higher Education, vol. 23(1), pp. 92-97, 2011.

[4] A. A. Sewagegn and B. M. Diale, "Empowering Learners Using Active Learning in Higher Education Institutions,' in Active Learning beyond the Future, London, U.K.: IntechOpen, 2019, ch. 3, pp. 31-41.

[5] Ribeiro MT, Singh S, Guestrin C., "Why Should I Trust You? Explaining the predictions of any classifier." In: Proceedings of the 22nd ACM SIGKDD international conference of knowledge discovery and data mining. 2016, pp. 1135–44. http://dx.doi.org/10.1145/2939672.2939778.

[6] Lundberg SM, Lee S., "A unified approach to interpreting model predictions." In: Proceedings of the 31st international conference on neural information processing systems (NIPS'17). Hook, NY, USA: Curran Associates Inc. Red; 2017, pp. 4768–77. http://dx.doi.org/10.5555/3294996.

[7] Alzubaidi, L., Zhang, J., Humaidi, A.J. et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," J Big Data 8, 53 (2021). https://doi.org/10.1186/s40537-021-00444-8

[8] D. Smilkov, S. Carter, A. Karpathy, C. Olah, D. Sculley, F, Viegas and M. Wattenberg, "The TensorFlow Playground", https://www.playground.tensorflow.org

[9] Gulsum Alicioglu, Bo Sun, "A survey of visual analytics for Explainable Artificial Intelligence methods," Computers & Graphics, vol. 102, 2022, pp. 502-520, ISSN 0097-8493, https://doi.org/10.1016/j.cag.2021.09.002.

[10] W. Aigner and F. Bodria and S. Rinzivillo and D. Fadda and R. Guidotti and F. Giannotti and D. Pedreschi., "Explaining Black Box with Visual Exploration of Latent Space," Eurographics Conference on Visualization, 2022. https://api.semanticscholar.org/CorpusID:252587516.

[11] N. Jethani, M. Sudarshan, I. Covert, S. Lee and R. Ranganath, "FastSHAP: Real-Time Shapley Value Estimation," eprint 2107.07436, in arXiv, 2022.

[12] M. Jdaitawi "The Effect of Flipped Classroom Strategy on Students Learning Outcomes," International Journal of Instruction, vol, 12 n3 pp. 665-680, Jul 2019

[13] M. Naik, "Assessing the Effectiveness of Flipped Classroom Strategy on Student Performance," European Chemical Bulletin. vol. 12(8). pp. 2883-2896, Aug. 2023.

[14] Mingorance Estrada, Á. C., Granda Vera, J., Rojas Ruiz, G., & Alemany Arrebola, I. (2019), "Flipped Classroom to Improve University Student Centered Learning and Academic Performance," Social Sciences, 8(11), 315. https://doi.org/10.3390/socsci8110315

[15] Z. Zainuddin and C. J. Perera, "Exploring students' competence, autonomy and relatedness in the flipped classroom pedagogical model," Journal of Further and Higher Education, vol. 43(1), pp. 115-126, Aug. 2017. DOI: 10.1080/0309877X.2017.1356916