VisionaryLLM: An Extensible Framework for Enhancing Large Language Models with Doman-Specific Vision Tasks Using Deep Learning

Hu Silan
School of Computing
National University of Singapore
e1373455@u.nus.edu

Tan Kah Xuan
School of Computing
National University of Singapore
e0692305@u.nus.edu

Abstract

While Large Language Models (LLMs) have demonstrated prominent capabilities, integrating them with domain-specific vision tasks remains challenging. We present VisionaryLLM, a framework that bridges this gap by providing a systematic approach to combining vision models with LLMs. Our framework addresses three key challenges: standardized integration of vision models, consistent interpretation of visual analysis, and extensibility across different domains. To validate our approach, we implemented the framework in two distinct domains: structural engineering for crack detection and medical imaging for breast ultrasound analysis. The framework's modular design allows seamless integration of both supervised and unsupervised learning approaches, demonstrating its flexibility in handling different types of vision tasks. Through our case studies, we show how VisionaryLLM facilitates natural language interaction with vision models while maintaining technical accuracy. The framework incorporates gradient-weighted class activation mapping (Grad-CAM) for result visualization, providing transparent insights into model decisions. Performance evaluation across different tasks demonstrates both the framework's effectiveness in maintaining model accuracy and its ability to adapt to different domains.

1 Introduction

1.1 Background and Motivation

General-purpose LLMs, such as ChatGPT, have recently emerged as a disruptive technologies with the potential to impact various aspects of society. Consequently, the use of these general-purpose LLMs has attracted considerable attention across diverse domains such as medicine, education, healthcare, finance, and academic writing. While this technology has the potential to transform these fields, its adoption is still accompanied by notable limitations and challenges.

One of the important applications of LLMs often lies among enterprises to improve work efficiency and reduce human costs. However, due to concerns around data privacy, intellectual property protection, and other security considerations, many enterprises prefer to deploy large models in local environments. For this reason, general-purpose LLMs often lack this domain-specific knowledge due to private proprietary data held within the organization. Furthermore, there is a lack of transparency in the model output due to the black-box nature of the deep learning models. Therefore, addressing these challenges provides a strong motivation for research opportunities, particularly in developing methods that enhance model interpretability and interactivity, leveraging LLMs and deep learning approaches to extract domain-specific knowledge in a transparent and accessible manner.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

1.2 Problem Statement

Our research identified three fundamental limitations in current LLM applications. **Firstly**, while LLMs can process and interpret images, they often lack the ability to provide domain-specific analysis. Such applications typically demand high reliability and precision, extending beyond basic detection or description. **Secondly**, the lack of visual explanation mechanisms in LLMs poses a significant challenge. When LLMs analyze images, they generate conclusions without providing insight into the underlying reasoning process or highlighting the specific visual features that informed their decisions. This "black-box" nature hinders users from verifying the analysis and comprehending how the conclusions were reached—an essential aspect in professional and high-stakes domains where transparency and accountability are crucial. **Lastly**, integrating LLMs with domain-specific visual tasks currently demands Considerable customization for each field, limiting their practicality for widespread professional adoption. A systematic, unified approach is necessary to facilitate consistent and cost-effective integration across diverse domains.

1.3 Project Objectives

Based on these observations, we developed VisionaryLLM with three main objectives:

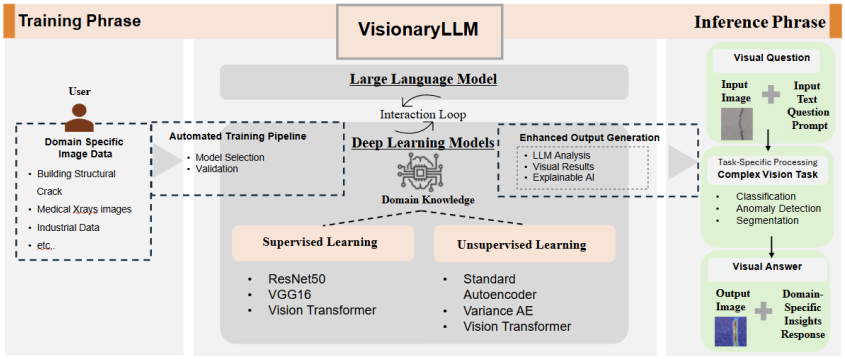

Figure 1: Imagination of VisionaryLLM

- To develop a framework that enhances LLMs with domain-specific visual analysis capabilities, ensuring both precision and natural language interaction
- To provide clear explanations of model decisions, helping users understand and verify the analysis results
- To design a simple integration approach that works effectively across different professional domains

Through this study, we aim to showcase the framework's efficacy across a range of complex vision tasks and demonstrate its versatility for various real-world applications. Figure 1 shows the overall framework of VisionaryLLM. This research could contribute to advancing the field of multimodal AI systems, offering valuable insights into the integration of vision and language models in practical settings.

1.4 Scope of the study

In this study, we explore the application of our framework across different domains by utilizing two distinct datasets: the "Concrete Crack Images for Classification" dataset, published by Özgenel et al. [6], which falls under the domain of structural monitoring, and the Beast ultrasound image dataset[13], which is relevant to the medical field. These datasets enable us to demonstrate the application of our framework in handling data from different domains, specifically engineering and healthcare.

2 Literature Review

2.1 Applications of LLMs in Domain Specific Vision Tasks

The advancement of Transformer architectures has enabled LLMs to demonstrate remarkable capabilities in natural language processing tasks, as explored in the studies by Min et. al. [5] and Zhang et. al.[14]. Recent works, such as Visual ChatGPT and HuggingGPT, have demonstrated the potential of LLMs to revolutionize traditional vision tasks through natural language interaction [12, 8]. These developments are particularly significant in professional domains like medical imaging and structural engineering, where efficient and intuitive visual analysis tools are essential. However, these general-purpose integrations face notable limitations in professional applications, primarily due to their lack of domain-specific expertise and inability to provide quantifiable and verifiable analytical results [4].

2.2 Effective Visual Analysis in Professional Domains

To address the specific requirements of professional visual analysis, two approaches have shown particular effectiveness. In scenarios with abundant labeled data, Transformer-based models have demonstrated superior performance due to their global attention mechanism[1]. Wang et. al. [11] and Guo et. al [2] respectively demonstrated these models' advantages in detecting complex patterns and details, which is crucial for structural defect analysis. In situations with limited labeled data, which is common in many professional applications, unsupervised methods such as Convolutional Variational Autoencoders (CVAE) have proven highly effective. Zhang et. al. [15] showed that CVAE achieved enhanced accuracy and noise resistance in structural analysis, while studies by Tsai et. al. [10] and Lee et. al. [3] demonstrated detection accuracy exceeding 90% across various professional applications.

2.3 Interpretability of Vision Models

Beyond accuracy, the interpretability of vision model decisions is crucial in professional applications. Grad-CAM has emerged as a powerful technique for visualizing the decision-making process of deep neural networks [7]. By highlighting the regions in input images that are most significant for predictions, Grad-CAM provides the necessary transparency for professional applications requiring model decision understanding and validation. This visualization capability is particularly valuable when integrating traditional vision models with LLMs, as it enables the generation of more accurate and verifiable explanations.

The success of these professional vision approaches, combined with the natural language capabilities of LLMs and the interpretability provided by techniques like Grad-CAM, indicates a promising direction for developing integrated solutions. Our VisionaryLLM framework builds upon these foundations to create a comprehensive solution that maintains the accuracy of professional visual analysis while providing intuitive natural language interaction and transparent decision processes.

3 Methodology

3.1 System Architecture

VisionaryLLM introduces a user-friendly framework that automatically enhances LLM capabilities with domain-specific vision tasks, as illustrated in Figure 2. Users need to provide domain-specific datasets (e.g., medical images or industrial data) through an interface. The framework then automatically handles the entire pipeline, from model selection to training and evaluation. The training stage features an automated pipeline that selects appropriate models, executes training processes, and evolves model performance. In the inference stage, the framework seamlessly integrates trained vision models with LLMs, providing both visual analysis and natural language explanations.

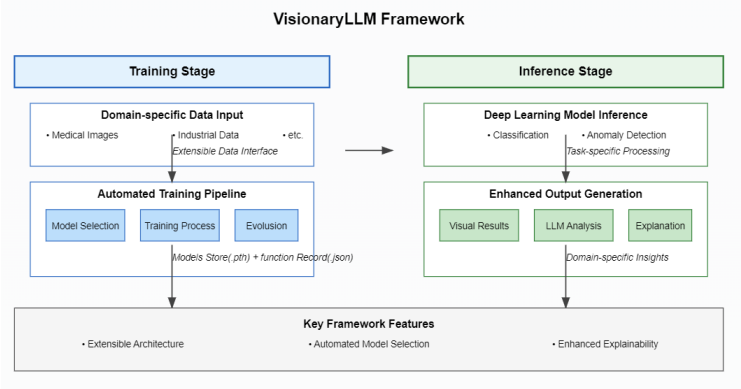

Figure 2: Usage Overview of VisionaryLLM Framework

3.2 Supervised Learning - Classification Model

3.2.1 Convolutional Neural Network

In this study, we have used some of the widely recognized pre-trained networks to simplify the application of CNNs in the classification task in images. Some examples include ResNet50 developed by Microsoft, VGG16 from the Oxford Visual Geometry Group, AlexNet created by Krizhevsky et. al. and EfficientNet. These models serve as foundational tools for transferring knowledge to various new tasks, streamlining their implementation.

3.2.2 Vision Transformer

In model design, we follow the base version of the ViTs presented by Dosovitskiy et. al. [1] which contains 12 layers with 16 by 16 input patch size. An overview of the model is shown in Figure 3. The ViTs is an extension from the standard Transformer to perform image classification tasks. In general, it consists of an embedding layer, an encoder, and a classifier as the final output layer.

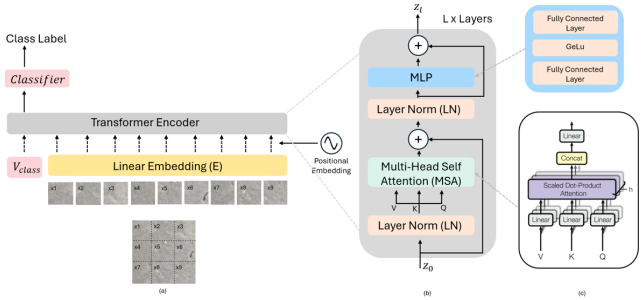

Figure 3: Vision Transformer Architecture: (a) Model's Main Architecture; (b) Transformer Encoder; (c) Multi-head Self Attention

The objective of ViTs is to learn how to map a sequence of image patches to their corresponding semantic label. Let $S = \{X_i, y_i\}_{i=1}^{r}$ denote the dataset of crack images, where X_i is an image , and y_i is its correponding label. Here, y_i is a binary class. In the first step, an image X is divided into non-overlapping patches. Each patch is treated as a distinct token by the Transformer. Each patch is linearly projected into a vector of dimension d using a learned embedding matrix E. The embedded representations are then concatenated with a learnable classification token v_{class}, used to perform the classification task. The resulting embedded sequence, z_0 is then given by $Z_0 = [v_{class}; x_1E; x_2E, ..., x_nE] + E_{pos}$, where $E \in \mathbb{R}^{(p^2c)\times d}$, $E_{pos} \in \mathbb{R}^{(n+1)\times d}$

In this study, the standard learnable 1-D position embedding is used. The sequence of embedded patches, z_0, is then fed into the Transformer encoder that consists of two main blocks:

- Multihead self-attention block (MSA) is denoted by $z'_l = MSA(LN(z_{l-1})) + z_{l-1}, wherel = 1, ..., L$
- Fully Connected Feed-forwards dense block (MLP) is denoted by $z_l = MLP(LN(z'_l)) + z'_l, wherel = 1, ..., L$

In the final layer of the encoder, the first element of the sequence z_L^0 is extracted and fed into a classifier to predict the class label,$y = LN(z_L^0)$

3.3 Unsupervised Learning - Anomaly Detection Model

3.3.1 Convolutional Autoencoder

To perform anomaly detection in image data, we employed both convolutional autoencoder (CAE) and convolutional variational autoencoder (CVAE) to learn feature representations of the input images in an unsupervised manner. Both architectures consist of two main components: an encoder and a decoder. The encoder extracts meaningful features from the input, and the decoder reconstructs the input based on these learned features. The representations are compressed into a latent space where the most important features are retained. For CAE, the hidden layers are composed of multiple convolutional layers that enable the effective encoding of meaningful features from image data. In addition to the CAE, we also utilized a CVAE that includes a probabilistic latent space instead of a fixed set of feature representations in CAE. An overview of the CAE is shown in figure 4 (a) and the overview CVAE is shown in figure 4 (b).

To evaluate the reconstruction quality, the error for each pixel was calculated by summing the squared differences between the original input and the CAE's reconstructed output across the three RGB channels. This reconstruction error, denoted as e can be expressed as: $e = \sum_{c=1}^{3}(p(x, y) - \hat{p}(x, y))^2$

where p and $\hat{p}$ denote the pixels of the original input image and the reconstructed output image, located at row x and column y, and c refers to the image color channel. By aggregating the reconstruction error across all pixels, an anomaly map of the image was generated, which was then used for defect detection.

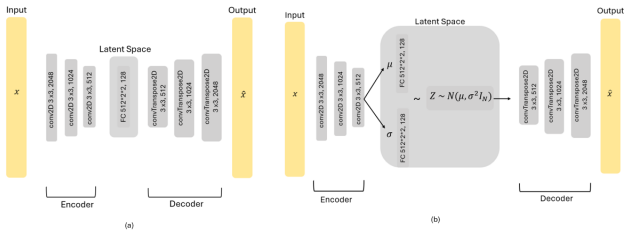

Figure 4: Unsupervised Learning - Crack Anomaly Detection Model (a) Convolutional Autoencoder (CAE); (b) Convolutional Variational Autoencoder (CVAE);

3.3.2 Model Traininig

For the supervised classification task, we divided the dataset into training and testing sets, allocating 20% of the data as the testing set to evaluate model performance. In the unsupervised crack anomaly detection task, we assumed a limited presence of cracked images in the dataset. Hence, we used 80% of the negative (non-crack) images and 10% of the positive (crack) images for training, ensuring a balanced approach that minimizes bias while allowing the model to detect rare anomalies effectively.

The models were trained with a batch size of 32 for 10 epochs, using a learning rate of 0.0001 and a weight decay of 0.0001. The AdamW optimizer was employed to optimize the training process.

3.4 Evaluation of Model Performance

To compare the performance of the various selected models, k-fold cross-validation will be used to compute the relevant performance metric of each model.

Table 1: Confusion matrix for binary classification

	Actual $y = 1$	Actual $y = 0$
Predicted $\hat{y_i} = 1$	True positive (TP)	False positive (FP)
Predicted $\hat{y_i} = 0$	False negative (FN)	True positive (TP)

In our study, we adopt the accuracy and F1 score as our main evaluation metrics which can be extracted by using a confusion matrix shown in Table 1, where $\hat{y_i}$ is the predicted class of the model i, and y is the true class. In particular: $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$, $Recall = \frac{TP}{FN+TP}$, $Precision = \frac{TP}{TP+FP}$ and $F1Score = \frac{2*Precision*Recall}{Precision+Recall}$.

3.5 Interpreting Model Decisions and Predictions

In addition to the evaluation metrics mentioned above, we employed the Gradient-weighted Class Activation Map (Grad-CAM) to generate heatmaps that emphasize the key regions of an image driving the model's predictions [7].

As described by Selvaraju et al. [7], the procedure to compute the class-discriminative localization map Grad-CAM, $L^c_{Grad-CAM} \in \mathbb{R}^{u \times v}$ of width u and height v for any class c is as follows. The gradient of the score for class c flowing into the final CNN layer is computed. It is y^c with respect to feature map activations A^k of a convolutional layer, $\frac{\partial y^c}{\partial A^k}$. The gradients flowing back are global-average pooled over the width and height dimension to beta the neuron importance weights, α^c_k as shown below:

$$\alpha^c_k = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}}$$

The importance weights, $\frac{\partial y^c}{\partial A^k}$, would indicates the importance of feature map k for a target class c. Then a weighted sum of the activations of the final CNN layer, $L^c_{Grad-CAM} \in \mathbb{R}^{u \times v}$, is calculated to obtain a coarse heatmap.

$$L^c_{Grad-CAM} = ReLU(\sum_k \alpha^c_k A^k)$$

The feature maps are then weighted using these gradients, creating a heatmap that highlights the key areas of the input image that holds significant contribution to the output class.

4 Experiment

We conducted two experiments to detect cracks in images, leveraging both supervised and unsupervised learning approaches. In the first experiment, a labeled dataset was used to train and test a classification model for crack detection (Section 4.1). The second experiment adopted an unsupervised learning approach, using the same dataset without labels to train the model to differentiate cracks (anomalies) from non-cracks (Section 4.2). Detailed results and analyses for each experiment are presented in the following subsections

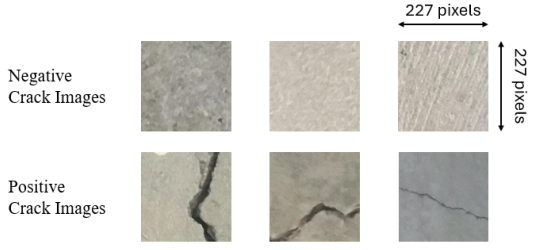

Figure 5: Sample images from each class

4.1 Experiment 1 : Classification Task

In this experiment, we used the dataset along with the provided labels. The results reported are based on evaluations conducted on the test dataset for the model selected in our study. Table 2 presents the performance metrics for the model trained from scratch without using pre-trained weights for 10 epoch.

Table 2: Comparison of Classification performance for Experiment 1 without using pre-trained weights. The best-performing model is highlighted in bold.

Model	Accuracy	Recall	Precision	F1 Score	Runtime (mins)
AlexNet	0.9980	0.9987	0.9972	0.9980	**13.14**
ResNet50	**0.9995**	**0.9998**	**0.9993**	**0.9995**	39.76
VGG16	0.9982	0.9975	0.9982	0.9979	77.59
ViT	0.9941	0.9990	0.9892	0.9941	120.48

As indicated in Table 2, all models achieved a similar F1 score of around 99%, with ResNet50 slightly outperforming the others. However, the ViT model was significantly less efficient in terms of computation time, requiring three times longer than ResNet50 to complete 10 epochs and reach comparable performance. This finding is further supported by Figure 6, which demonstrates that ResNet50 reaches a lower training loss faster than ViT.

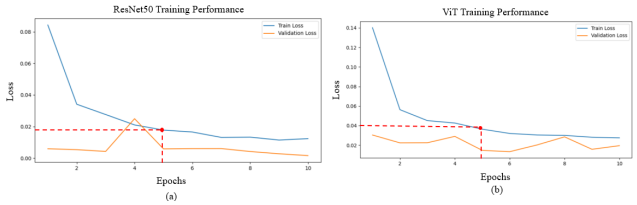

Figure 6: Training Performance (a) ResNet50 Loss per Epoch; (b) ViT Loss per Epoch;

4.2 Experiment 2 : Anomaly Detection Task

To address the common challenge of obtaining labeled dataset, we formulated the task to train the model to detect structural cracks in an unsupervised approach. the result of this experiment is shown in Table 3.

Table 3: Comparison of anomaly detection performance for Experiment 2. The best-performing model is highlighted in bold.

Model	Accuracy	Recall	Precision	F1 Score	Runtime (mins)
Standard AE	0.9557	0.5359	0.9696	0.6868	**13.91**
VAE	0.9418	0.3775	0.9557	0.5413	15.94
ViT Anomaly	**0.9968**	**0.9987**	**0.9947**	0.9967	22.48

It can be observed from Table 3 that as compared to the Standard AE, VAE has scored 14.55% lower in F1 score, Among the selected models, ViT Anomaly performed the best, scoring an F1 score of 99% on the test dataset.

4.3 Experiment 3: Complements LLM reflections on returning vision

To further analyze model interpretability, we applied Grad-CAM to each model's output. For example, in structural crack detection, Grad-CAM accurately pinpointed crack locations, underscoring the model's focus on key structural details as shown in figure 7. These insights from Grad-CAM not only validate the models' outputs but also offer additional transparency, allowing users to understand the rationale behind each model's predictions.

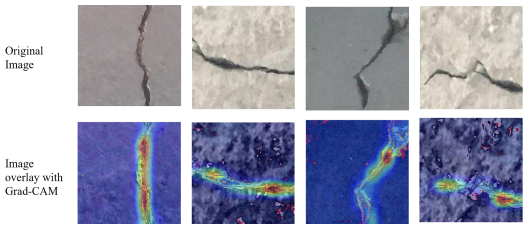

Figure 7: Sample image output using Grad-Cam

5 Cross-domain Application of VisionaryLLM

In this section, we demonstrate how our framework effectively translates across different domains as discussed in our methodology. Figure 8 showcases its application in medical imaging analysis while it is also applicable to detect building cracks in images. In this medical scenario, the user uploads a breast ultrasound image, along with a prompt: "Please help me analyze this tumor photo." The model, trained on the BreastMNIST dataset, indicates a 90.67% probability that the image contains a tumor, demonstrating its capability for probabilistic assessment similar to its application in structural defect analysis. The Grad-CAM visualization, comparable to our crack detection results shown in Figure 7, provides transparent insight into the model's decision-making process by highlighting the suspicious regions in the ultrasound image. This implementation highlights our framework's core components – visual analysis, natural language interaction, and interpretable results while maintaining their effectiveness across different domains. The user interface allows both engineers and medical professionals to make domain-specific inquiries while benefiting from the same underlying technical capabilities, validating our framework's design principle of domain adaptability while maintaining professional precision.

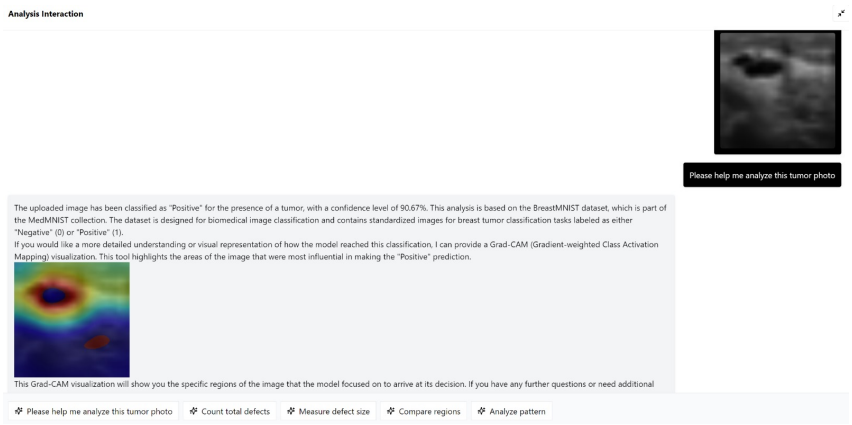

Figure 8: Interpretable Analysis of Breast Tumor Detection: Original medical image (top) and corresponding Grad-CAM Visualization (bottom), demonstrating model's region of interest with 90.67% confidence

6 Conclusion

In conclusion, the integration of domain-specific vision tasks with large language models (LLMs) through the VisionaryLLM framework demonstrates significant potential for enhancing model performance and user interaction across diverse domains. Our evaluation of models like ResNet50, VGG16, and Vision Transformer revealed that while ResNet50 performed best in tasks requiring local feature detection, Vision Transformer showed promise in capturing global context. The use of Grad-CAM further enhanced model interpretability, providing transparent insights into the decision-making process. Overall, VisionaryLLM not only improves the accuracy of domain-specific tasks but also fosters greater user confidence and interaction through its ability to offer detailed, understandable model outputs, highlighting the potential for further refinement and broader application in complex

real-world scenarios. Our repository [VisionaryLLM] (https://github.com/Qingbolan/Vision-LLM-Integration)[9]

References

[1] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[2] Feng Guo, Jian Liu, Chengshun Lv, and Huayang Yu. A novel transformer-based network with attention mechanism for automatic pavement crack detection. *Construction and Building Materials*, 391:131852, 2023.

[3] Kanghyeok Lee, Seunghoo Jeong, Sung-Han Sim, and Do Hyoung Shin. Damage-detection approach for bridges with multi-vehicle loads using convolutional autoencoder. *Sensors*, 22(5):1839, 2022.

[4] Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, et al. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*, 2023.

[5] Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40, 2023.

[6] Ç F Özgenel and A Gönenç Sorguç. Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In *Isarc. proceedings of the international symposium on automation and robotics in construction*, volume 35, pages 1–8. IAARC Publications, 2018.

[7] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[8] Y. Shen et al. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *arXiv preprint arXiv:2303.17580*, 2023.

[9] silan Hu. Visionaryllm: An extensible framework for enhancing large language models with doman-specific vision tasks using deep learning. `https://github.com/Qingbolan/Vision-LLM-Integration`, 2024.

[10] Du-Ming Tsai and Po-Hao Jen. Autoencoder-based anomaly detection for surface defect inspection. *Advanced Engineering Informatics*, 48:101272, 2021.

[11] Wenjun Wang and Chao Su. Automatic concrete crack segmentation model based on transformer. *Automation in Construction*, 139:104275, 2022.

[12] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023.

[13] Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 191–195, 2021.

[14] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[15] Yonglai Zhang, Xiongyao Xie, Hongqiao Li, and Biao Zhou. An unsupervised tunnel damage identification method based on convolutional variational auto-encoder and wavelet packet analysis. *Sensors*, 22(6):2412, 2022.