A transformer-based model to predict the occupancy rate of public transportation buses at bus stops

1st Bruno Rocha Toffoli
Software Development Department
Geocontrol SA
Vitoria, Espirito Santo, Brazil
btoffoli@gmail.com

2nd Eduardo Lima Pereira
Department of Industrial Automation
Federal Institute of Espírito Santo
Vitoria, Espirito Santo, Brazil
eduardo.pereira@ifes.edu.br

3rd Igor Aguiar Rodrigues
Software Development Department
Geocontrol SA
Vitoria, Espirito Santo, Brazil
igor_aguiar@yahoo.com.br

***Abstract*—Accurate prediction of bus occupancy rates at specific stops is crucial for optimizing public transportation systems, enhancing commuter satisfaction, and improving operational efficiency. This paper presents a deep learning approach using a Transformer based model to predict bus occupancy levels. The study leverages a comprehensive dataset provided by Geocontrol, detailing trip information from the TRANSCOL public transport system in Grande Vitoria, Brazil. This dataset includes diverse features such as temporal records, route specifications, passenger flow metrics, and stop level details, with occupancy classified into three distinct levels derived from onboard image analysis. The proposed model utilizes an encoder only Transformer architecture designed to handle the combination of numerical and categorical input features extracted via a robust preprocessing pipeline. The model aims to predict the sequential occupancy levels for each stop within a bus trip. Through this research, we seek to contribute to the advancement of intelligent public transport systems by applying state of the art deep learning methodologies for more accurate occupancy forecasting, ultimately supporting better resource planning and service reliability. Initial findings indicate the model's potential while also suggesting directions for future refinement.**

***Index Terms*—bus occupancy, transformer**

I. Introduction

Public transportation is vital in densely populated urban environments, significantly impacting commuter satisfaction and overall service efficiency. One of the key metrics influencing the performance of public transit systems is the occupancy rate of buses at various stops. Accurate prediction of occupancy levels can improve operational planning, optimize fleet utilization, and improve service reliability.

This study addresses these challenges by proposing a Transformer-based model to predict occupancy levels in a public transportation system. This research uses a comprehensive dataset provided by Geocontrol, which includes detailed trip information from the TRANSCOL public transport system in Grande Vitória, Brazil. The dataset encompasses temporal records, route and vehicle specifications, passenger flow metrics, operational details, service interruptions, and stop-level information. Given the diverse nature of these attributes, an advanced deep learning approach is implied to extract meaningful patterns and improve prediction accuracy.

To this end, a transformer-based model has been designed and implemented that classifies occupancy levels into three distinct categories (0, 1, or 2) based on a combination of numerical and categorical features. The model leverages an encoder-only Transformer architecture, which is an option for handling structured sequential data across various domains. The implementation processes large datasets through incremental learning techniques, ensuring scalability and robustness in real-world scenarios.

Some objectives of this study also include developing a robust preprocessing pipeline to manage mixed data types efficiently and provide a well-justified architectural and implementation framework for future research and practical deployment.

Through this research, we aim to contribute to the ongoing advancement of intelligent public transport systems by leveraging state-of-the-art deep learning methodologies.

II. Related Works

The analysis of passenger occupancy in vehicles and passenger load on network segments has been a critical aspect of public transportation operations, aiding in service optimization and passenger demand adaptation. Traditionally, surveyors manually counted passengers at key transit locations. However, with the advent of Intelligent Transportation Systems (ITS), data collection has been streamlined through Automated Passenger Counting (APC) systems and various sensor-based technologies, such as cameras, LiDAR, weight sensors, and Wi-Fi tracking [1], [2]. Some of these technologies enable real-time transmission, allowing the development of predictive models for passenger occupancy and demand forecasting.

Effective prediction of passenger occupancy is crucial for both travelers and service operators. Various forecasting approaches have been explored, ranging from simple historical average models to sophisticated machine learning techniques [3]. Traditional statistical models, such as linear regression, Poisson regression, and autoregressive integrated moving average (ARIMA), have been commonly used due to their interpretability and moderate accuracy [4], [5]. However, these models often fail to capture non-linear dependencies and external influencing factors, such as weather and special events.

To address these limitations, machine learning techniques have been increasingly adopted. Support Vector Regression (SVR) is one such method, capable of handling non-linear

relations and defining acceptable error margins [**?**]. Ensemble learning methods, such as Random Forest (RF) and XGBoost, have also demonstrated promising results in predicting passenger demand with moderate data requirements and high scalability [6].

Neural networks, particularly deep learning models, have gained popularity due to their ability to capture complex temporal-spatial patterns. Long Short-Term Memory (LSTM) networks have shown strong predictive performance in time-series analysis, especially in dealing with recurrent transportation patterns [7], [8]. Additionally, hybrid models integrating multiple approaches, such as combining RF with LSTM, have been proposed to improve prediction accuracy and robustness [9]. More recently, Graph Neural Networks (GNNs) have been applied to passenger occupancy forecasting, leveraging the network-like structure of public transportation systems to enhance prediction accuracy [10].

Transformer-based models [11] have achieved unparalleled performances in many long-standing AI tasks in natural language processing and computer vision fields, thanks to the effectiveness of the multi-head self-attention mechanism. This has also triggered lots of research interest in Transformer-based time series modeling techniques. In particular, a large amount of research works are dedicated to the LTSF task. Considering the ability to capture long-range dependencies with Transformer models, most of them focus on the less-explored long-term forecasting problem

The core structure of the Transformer consists of an encoder-decoder framework, where each component is composed of multiple layers of self-attention mechanisms and feedforward networks [12]. This hierarchical design allows the model to capture intricate contextual relationships within input sequences, thereby enhancing its ability to process large datasets effectively. One of the pivotal components of the Transformer model is the multi-head attention mechanism, which enables each element in an input sequence to learn relationships with other elements from multiple perspectives simultaneously [13]. This capability facilitates learning complex patterns within the data, making the Transformer architecture highly adaptable across diverse applications. Furthermore, positional encoding techniques are incorporated to help the model retain the sequential information of data, compensating for the lack of inherent recurrence in the structure.

The Transformer architecture has served as the foundation for numerous advanced models, such as BERT and GPT, both of which have demonstrated state-of-the-art performance in NLP tasks [14], [15]. Pretrained Transformer models have been particularly effective in transfer learning scenarios, enabling significant improvements across various natural language understanding benchmarks. Devlin et al. (2019) illustrated the superior performance of BERT in tasks such as sentiment analysis, question answering, and named entity recognition [15]. The applicability of Transformers extends beyond NLP, with research demonstrating their effectiveness in domains such as image processing. For instance, Dosovitskiy et al. (2021) introduced the Vision Transformer (ViT), which applies self-attention mechanisms to image patches, outperforming traditional convolutional neural networks (CNNs) in large-scale image classification tasks [16].

Beyond NLP and computer vision, Transformer models have been successfully applied in other fields, including transportation prediction systems. The ability of Transformers to capture complex temporal dependencies and process long time-series data makes them particularly effective in modeling intricate transportation patterns [17]. This broad applicability underscores the transformative impact of the Transformer architecture in modern machine learning applications and highlights its potential for future research and industrial applications.

In summary, while traditional models provide a baseline for passenger occupancy prediction, machine learning and deep learning approaches offer enhanced accuracy and adaptability. Future research should improve model interpretability and reduce computational costs to facilitate real-time implementation in ITS environments.

III. Data preprocessing

A. Data acquisition

In this study, we utilize a comprehensive dataset provided by Geocontrol, encompassing detailed trip information from the Transcol public transport system in Grande Vitória, Brazil. The dataset, collected over 1 year, includes extensive attributes related to each trip, such as scheduled versus actual start and end times of trips, route details, bus identification, number of stops, route distance, trip duration, and periodic records of the level of occupancy of the bus. The occupancy level of a bus is quantified using a neural network model that processes internal images captured at regular intervals during each trip. The model classifies the occupancy into three distinct levels:

- Level 0: Empty or nearly empty;
- Level 1: Half full – seating is unavailable, but standing space remains;
- Level 2: Full – neither seating nor standing space is available.

This classification not only informs about current capacity utilization but also serves as a foundational metric for predicting service performance and planning resource allocation.

B. Feature engineering

This section outlines the data processing pipeline employed to transform raw trip records into a structured, model-ready format. The methodology encompasses feature engineering techniques and tokenization processes essential for enhancing predictive model performance. The feature engineering phase begins with an intricate analysis of the data. We decompose dates into their constituent days and apply a sinusoidal transformation to capture the cyclical nature of days and weeks. One One-hot encoding is applied to convert them into a format suitable for machine learning algorithms. The methodology involves multiple steps to process the raw input data effectively. Raw timestamps, originally in ISO format, are converted into timezone-aware datetime objects to ensure consistency across

different time zones. To better capture periodic patterns in the data, sine and cosine transformations are applied to the hour-of-day and day-of-week features. The trip delay is computed in minutes by comparing the scheduled and actual trip times. Delays are then categorized based on predefined thresholds to facilitate classification. Bus stop locations are normalized by dividing their position by the total route length, ensuring standardized spatial representation. Continuous weather attributes, such as temperature and precipitation, are discretized into categorical variables to improve model interpretability and performance. Additional contextual features, including month, bimester, trip route identification, and estimated occupancy levels, are extracted to enrich the dataset.

A neural network model is employed to estimate bus occupancy levels based on internal images captured at regular intervals during trips. The model classifies occupancy into three distinct levels: Level 1, indicating an empty or nearly empty bus with ample seating available; Level 2, representing a half-full bus where no seating is available, but standing space remains; and Level 3, denoting a full bus with neither seating nor standing space available. This classification provides critical insights into passenger load dynamics, facilitating optimized public transport operations and improved resource allocation. The methodology ensures a structured approach to processing raw transport data, enabling accurate and efficient passenger occupancy prediction.

To make the structured data suitable for a language model, each data record was transformed into a natural language prompt. This process, known as prompt engineering, frames the prediction task as a question-answering or instruction-following task for the LLM.

A consistent template was designed to format the input features into a coherent textual prompt. An example template is:

```
"Given the following conditions for a bus
journey:
Day Type: [Day]
Precipitation: [Precipitation Status]
Temperature: [Temperature Status]
Route Number: [Route Number]
Scheduled Start Time: [Scheduled Time]
Actual Start Time: [Actual Start Time]
Stop Number: [Stop Number]

Predict the occupancy level for this
specific stop."
```

IV. Fine-tunning approach

This section details the methodology employed to predict bus occupancy levels using a fine-tuned Large Language Model (LLM). The process involves model selection, fine-tuning using the Unsloth library for efficiency, and evaluation strategy.

A. Model selection

The base model selected for fine-tuning was Mistral-7B (or a specific variant like mistralai/Mistral-7B-Instruct-v0.2) [18]. Mistral models are known for their strong performance relative to their size, making them suitable candidates for fine-tuning on specific tasks. The instruction-tuned variant (Instruct) is particularly well-suited as it is already trained to follow instructions presented in prompts.

B. Fine-tuning

To optimize the fine-tuning process in terms of computational resources (GPU memory) and speed, the Unsloth library was employed. Unsloth provides highly optimized implementations of training techniques, including memory efficiency, speed enhancements, and parameter-efficient fine-tuning (PEFT): Unsloth integrates seamlessly with PEFT methods like QLoRA (Quantized Low-Rank Adaptation), which involves quantizing the base model (e.g., to 4-bit precision) to reduce memory usage and then training small.

The model architecture is based on the Transformer neural network, specifically an encoder-only configuration. The Transformer, relies entirely on self-attention mechanisms and has been widely adopted due to its ability to capture long-range dependencies. An encoder-only Transformer is chosen for this many-to-one classification task, where multiple input features map to a single occupancy level output. This architecture is computationally efficient and well-suited for feature extraction.

The model implementation is built using PyTorch's Transformer Encoder and Transformer encoder layer classes, which provide optimized implementations of multi-head self-attention, feed-forward layers, residual connections, and layer normalization. Notably, positional encoding is excluded from the architecture because the input features do not represent a sequence with a meaningful order, unlike text or time series data. Instead, they are independent features describing a single event, such as a bus stop visit. Since no sequential dependency exists, positional encoding is unnecessary. Furthermore, the batch processing of events in this model does not rely on any inherent order among samples, reinforcing the decision to omit positional encodings.

This methodological framework integrates data preprocessing, feature engineering, and a robust Transformer-based classification model to achieve reliable bus occupancy estimation. The structured approach enhances model interpretability and predictive performance, supporting more efficient public transportation management.

V. Full training approach

The model is trained on a small subset of the dataset to demonstrate its effectiveness. It uses only data from a single route over one month, avoiding the complexity of multi-route or multimonth data. For example, for dealing with 1000 routes, it would be required an embedding layer to encode the route information. In this sense, the features extracted from the raw data are not exhaustive, and additional features could be included to enhance model performance.

The dataset consists of JSONL files containing trip records from a single route over one month. Each record includes features such as timestamp, weather conditions, and bus stop location. The dataset is pre-split into training (80%), validation (10%), and test (10%) sets:

- Training: 633 trips, 38,762 records
- Validation: 78 trips, 4,495 records
- Test: 78 trips, 5,002 records

The dataset exhibits class imbalance: Empty (0): 17,787 samples (37.9%); Half Full (1): 22,718 samples (48.4%); and Full (2): 7,754 samples (16.7%). To address this imbalance, class weights are applied during training, using 0.90 for Empty, 0.71 for Half Full, and 2.07 for Full.

The model follows a transformer-based architecture comprising the following components:

A. Input Embedding

The input features are passed through a linear layer to project them into a higher-dimensional embedding space (`hidden_dim=1024`). This transformation ensures that the model can capture complex relationships between features.

B. Positional Encoding

To incorporate sequence order information into the embeddings, positional encoding is applied. This step is crucial for transformers, which lack inherent sequential awareness. The encoding is computed using sine and cosine functions:

$$\text{pe}_{[:,0::2]} = \sin(\text{position} \times \text{div\_term}) \tag{1}$$

$$\text{pe}_{[:,1::2]} = \cos(\text{position} \times \text{div\_term}) \tag{2}$$

C. Transformer Encoder

The core of the model consists of a stack of transformer encoder layers (`num_encoder_layers=6`) with multi-head self-attention (`nhead=8`). Each layer processes the input sequence in parallel, capturing dependencies across all positions.

D. Classification Head

A two-layer multilayer perceptron (MLP) with ReLU activation and dropout serves as the classification head. It outputs logits for each occupancy level.

E. Justification

- **Transformers**: These architectures excel at modeling sequential data and capturing long-range dependencies, making them ideal for trip-based predictions.
- **Positional Encoding**: Ensures the model understands the temporal order of bus stops.
- **Class Weights**: Mitigates the impact of class imbalance by emphasizing minority classes during training.
- **Sequence-to-Sequence Prediction**: The model predicts the occupancy level for each bus stop in a trip, enabling detailed insights into occupancy patterns.

VI. Model Output

The model predicts the occupancy level for each bus stop in a trip. Specifically:

- The input to the model is a sequence of features corresponding to all bus stops in a single trip.
- The output is a sequence of predictions, where each prediction corresponds to the occupancy level (`0`, `1`, or `2`) at each bus stop in the trip.

Thus, the model performs sequence-to-sequence prediction, meaning it takes a sequence of inputs (features for all bus stops in a trip) and outputs a sequence of predictions (occupancy levels for all bus stops).

VII. Results

A. Fine-tunning approach results

The Mistral-7B model consistently excelled in generating JSON-formatted responses, frequently incorporating comprehensive metadata. It demonstrated the most structured and predictable output format among the models tested, showcasing its ability to adapt effectively to complex instruction sets. However, several challenges were identified during the fine-tuning process. One of the primary issues was inconsistent response formats, where the model often deviated from the straightforward numeric occupancy level predictions (e.g., 0, 1, 2). Additionally, there was a tendency to include excessive metadata, which detracted from the focus on the core prediction task. Another notable challenge was the ambiguity in interpretation, as the model sometimes produced responses such as ”100%” instead of providing precise occupancy level predictions. From a technical perspective, the implementation relied on the original technical stack. The framework used for the development and fine-tuning processes was PyTorch, with fine-tuning optimization managed through Unsloth. HuggingFace was employed for model management, and the hardware setup included an Intel i7 processor, 32GB RAM, and an NVIDIA RTX 3060 GPU. This robust setup facilitated the experimentation and analysis of the model's performance across varying tasks.

B. Full training approach results

This approach's results are presented in 2, 4, 1, and 3.

The figures show that the model is not ready to be used in a commercial way, but it is a beginning for new challenges.

Acknowledgment

The authors would like to express their sincere gratitude to GeoControl for their invaluable support and contributions to this work. Their resourceshave greatly enhanced the quality of this research. Additionally, the authors are deeply thankful to Professor Alberto for his insightful guidance and encouragement throughout the development of this study. This accomplishment would not have been possible without their significant input and dedication.

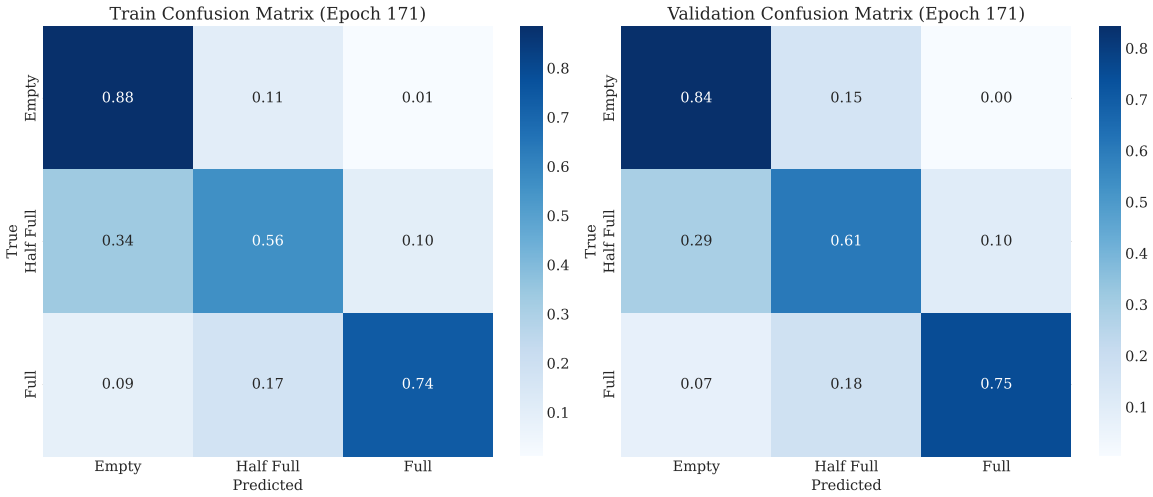

Fig. 1. Confusion Matrices

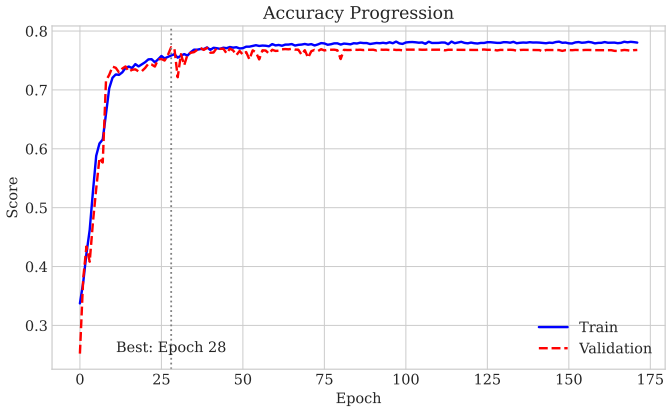

Fig. 2. Accuracy Progression

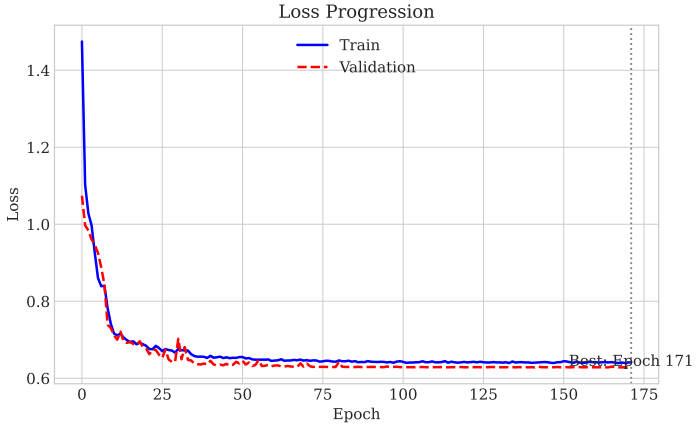

Fig. 3. Loss Progression

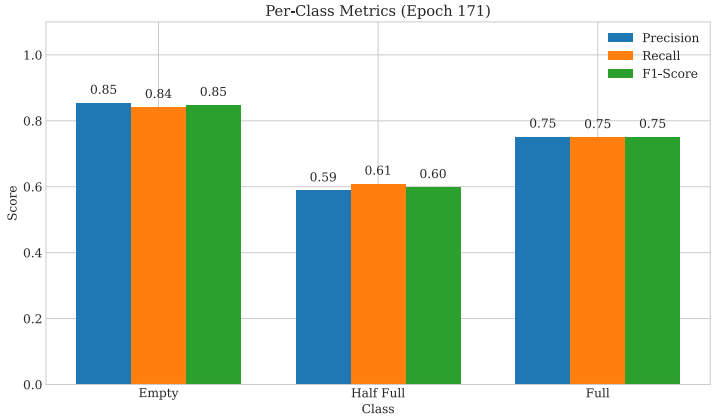

Fig. 4. Class Metrics

REFERENCES

[1] C. Mccarthy, I. Moser, P. P. Jayaraman, H. Ghaderi, A. M. Tan, A. Yavari, U. Mehmood, M. Simmons, Y. Weizman, D. Georgakopoulos *et al.*, "A field study of internet of things-based solutions for automatic passenger counting," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 2, pp. 384–401, 2021.

[2] L. Zou, S. Shu, X. Lin, K. Lin, J. Zhu, and L. Li, "Passenger flow prediction using smart card data from connected bus system based on interpretable xgboost," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, p. 5872225, 2022.

[3] J. Wood, Z. Yu, and V. V. Gayah, "Development and evaluation of frameworks for real-time bus passenger occupancy prediction," *International Journal of Transportation Science and Technology*, vol. 12, no. 2, pp. 399–413, 2023.

[4] E. Jenelius, "Personalized predictive public transport crowding information with automated data sources," *Transp. Res. C, Emerg. Technol.*, vol. 117, 2020.

[5] C. Xu, Z. Li, and W. Wang, "Short-term traffic flow prediction using a methodology based on autoregressive integrated moving average and genetic programming," *Transport*, vol. 31, no. 3, pp. 343–358, 2016.

[6] F. C. F. Gallo and N. Sacco, "Real-time occupancy predictions of public transport vehicles," *Swiss Transp. Res. Conf. (STRC)*, 2022.

[7] K. Pasini, M. Khouadjia, A. Same, F. Ganansia, and L. Oukhellou, "Lstm encoder-predictor for short-term train load forecasting," in *Joint*

European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 2019, pp. 535–551.

[8] L. Monje, R. A. Carrasco, C. Rosado, and M. Sánchez-Montañés, "Deep learning xai for bus passenger forecasting: A use case in spain," *Mathematics*, vol. 10, no. 9, p. 1428, 2022.

[9] S. Lin and H. Tian, "Short-term metro passenger flow prediction based on random forest and lstm," *IEEE 4th Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, 2020.

[10] J. Z. et al., "Multi-graph convolutional network for short-term passenger flow forecasting in urban rail transit," *IET Intell. Transp. Syst.*, vol. 14, no. 10, pp. 1210–1217, 2020.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[12] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[13] K. Ahmed, B. C. Wallace, R. Johnson, M. T. Pilehvar, and Y. Wang, "Understanding attention mechanisms in text classification," pp. 1920–1930, 2019.

[14] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training."

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2019.

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, and S. Gelly, "An image is worth 16 × 16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[17] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, and F. S. Khan, "Transformers in vision: A survey," *arXiv preprint arXiv:2101.01169*, 2021.

[18] M. AI, "Mistral 7b: The best model to date, apache 2.0," https://mistral.ai/news/announcing-mistral-7b/, 2023, accessed: 2025-03-27.