EasyChair Preprint

№ XXX

VISION-RISK: Vision-Language Model for Risk Assessment in Explainable Autonomous Driving Systems

Andrei Bogdan Constantin, Sebastian-Antonio Toma, Vlad Negru, Camelia Lemnaru and Rodica Potolea

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

March 23, 2026

VISION-RISK: Vision-Language Model for Risk Assessment in Explainable Autonomous Driving Systems

1st Andrei-Bogdan Constantin
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
andrei.constantin042@gmail.com

2nd Sebastian-Antonio Toma
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
sebastianantoniotoma@gmail.com

3rd Vlad Andrei Negru
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
Vlad.Negru@campus.utcluj.ro

4th Lemnaru Camelia
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
Camelia.Lemnaru@campus.utcluj.ro

5th Rodica Potolea
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
Rodica.Potolea@campus.utcluj.ro

***Abstract*—The advancements of autonomous driving systems hinge on their ability to navigate complex environments while ensuring safety and transparency. The lack of explainability in the current technologies - the ability to provide clear, human-readable justifications for actions - undermines trust, complicates validation and hinders widespread adoption. In this paper, we introduce VISION-RISK, a vision-language model (VLM) designed for risk assessment and explainability in autonomous driving using a lightweight architecture, optimized for deployment on edge devices. To train the model, we developed a custom dataset combining real-world driving scenarios from Honda Driving Dataset (HDD) and extreme high-risk cases from Crash1500, augmented with synthetic annotations using Dolphins and refined via DeepSeek V3. VISION-RISK stands out through three key characteristics: the integration of danger level classification with natural language explanation generation, a lightweight architecture optimized for deployment on resource-constrained devices, and a strong emphasis on interpretability and safety to enhance trust in autonomous systems.**

I. Introduction

The pursuit of achieving full autonomy in driving has fueled innovation processes, combining knowledge from domains like artificial intelligence, robotics and automotive engineering. In essence, the ultimate goal is to develop autonomous driving vehicles, capable of navigating through complex real-world environments while demonstrating a level of understanding and adaptability comparable to that of a human driver.

In recent years, autonomous driving technologies have made a significant progress, more specifically, with the help of deep learning models that are capable of interpreting complex visual scenarios and making real-time decisions. Nevertheless, a major challenge persists: the lack of transparency and limited interpretability of the outputs in these decision-making processes as most current autonomous driving systems are data-driven, featuring end-to-end architectures that transform inputs into control output directly [1] [2].

To address these challenges, a growing number of auxiliary systems are being developed, that are not necessarily focused on direct vehicle control, but rather of improving the behavioral comprehension [3] [4] [5] [6] [7] [8]. Usually, these solutions are separated from the main decision-making process and are integrated as additional modules, complementing the core functionalities of the autonomous systems through textual explanations, offering an extra layer of transparency.

In this context, a crucial first step toward enhanced safety and building trust in these models lies in the essential need to explicitly assess the danger level of traffic situations and provide a textual explanation [9]. Such a mechanism would increase the transparency of the system's decisions, while adding an additional layer of safety, enabling both internal validation of the model's behavior and external understanding by human users.

In this paper, we propose VISION-RISK, a vision-language model (VLM) designed for explainability in autonomous driving, specifically, a scenario risk assessment model that processes video input and generates a risk level label (Low, Medium or High) and a corresponding textual explanation of that. To support this approach, we introduce:

1) VISION-RISK model: a lightweight multimodal architecture capable of processing driving video data and generating both risk assessments and natural language explanations
2) A custom dataset: built from the Honda Driving Dataset (HDD) and Crash1500, containing driving video scenarios, corresponding risk labels and textual explanations. These annotations guide the model's learning process in order to associate visual patterns with both the driver's risk exposure and the rationale behind each situation.

II. Related Work

A. Dolphins: A Multimodal Language Model for Autonomous Driving

Dolphins [10] is a visual-language model specifically tailored for autonomous driving domain and presented as a conversational driving assistant. It is built on OpenFlamingo's architecture (a multimodal language model that can be used for variety of tasks) [11] and supports complex driving-related reasoning by integrating video and textual instructions. Unlike conventional end-to-end autonomous systems or unimodal language-driven agents, Dolphins is capable of emulating human-like understanding of driving scenarios through in-context learning and multimodal perception.

To tackle challenges like lack of holistic understanding, Dolphins implements Ground Chain of Thought (GCoT) [12] in its approach. This method involves enhancing model's fine-grained reasoning capabilities by training on visual question answering tasks (VQA) with step-by-step explanations generated with ChatGPT [13]. These learned capabilities are transferred to driving-specific tasks through fine-tuning on BDD-X dataset [14].

There are four types of tasks relevant to autonomous systems that are handled by Dolphins: Behavior Understanding, Behavior Reasoning, Prediction with Control Signals, Detailed Conversations.

Benchmarks tests demonstrate that Dolphins has strong generalization across driving tasks and notable zero-shot and few-shot performances. Current limitations of the model are computational load and inference speed, which are planned to be addressed through future optimizations like knowledge distillation of the model.

B. Honda Research Institute Driving Dataset (HDD)

The Honda Research Institute Driving Dataset (HDD) [15] is a large-scale dataset designed to facilitate research in driving behavior understanding and causal reasoning. It consists of over 100 hours of real-world driving data, collected in the San Francisco Bay area.

Apart from the data collected from sensors (such as speed, turn angle, turning speed), a key innovation of HDD is in its 4-layer hierarchical annotation scheme. This structure, which we have leveraged in our dataset for VISION-RISK, provides a structured understanding of human driving behaviors in different scenarios. Each session is annotated across the following layers:

- Goal-oriented action (e.g. right turn, merge)
- Stimulus-driven Action (e.g. stop)
- Cause (e.g. sign, congestion)
- Attention (e.g. pedestrian, vehicles)

This multi-layer approach, enables the dataset to represent complex driver behavior with both actions and causes, allowing for study and research of multitask learning, real-time driver behavior prediction and interpretability in decision-making models.

C. HazardVLM: A VLM for Real-Time Hazard Detection in Autonomous Driving

HazardVLM [16] is a multimodal model developed to overcome the problem of hazard recognition and real-time short explanation in autonomous driving scenarios. The architecture integrates vision-language modeling and it is aimed to detect high-risk hazards in driving scenarios and provide a human-readable short textual justification. The key innovations of HazardVLM are:

- *Lightweight architecture* - avoids computational overhead of generic models by employing a domain specific language model
- *Adaptive Sampling for Efficiency* - dynamically samples frames
- *Interpretability Features* - integrates an activation mechanism (Eigen-CAM) to highlight spatio-temporal features
- *Novel Dataset (DoTA-HEC)* - introduces a real-world hazard caption dataset with structured annotations

The architecture follows an encoder-decoder framework:

- Encoder: 3D CNN that extracts spatio-temporal features from sampled frames
- Visual Tokenizer: It compresses high-dimensional features into compact tokens for efficient decoding
- Decoder: A transformer-based autoregressive model that generates hazard descriptions

While HazardVLM makes significant strides in combining vision-language understanding, its focus is primary on high-risk hazards, without offering a nuanced spectrum of risk levels and very complex textual explanations.

D. BDD-X: A Benchmark Dataset for Explainable Autonomous Driving

The Berkeley DeepDrive eXplanation dataset (BDD-X) [14] is a large-scale multimodal dataset designed to bridge the gap between autonomous vehicle decision-making and human-readable explanations. Expanding upon Berkeley DeepDrive dataset (BDD) [17], it enriches the dashcam video data with time-stamped textual annotations and justifications for driving behaviors, enabling research into explainable AI for self-driving systems.

BDD-X's annotations focus on routine driving maneuvers (e.g. turns, merges, accelerations) and lacks extreme-edge cases like collisions, which are essential for training robust models that take in account hazard predictions.

E. Spatio-temporal modeling in video analysis and multimodal models

Video data is multimodal and contains both spatial (individual frames) and temporal information (sequence of frames). In order to process this kind of data, models capable of learning and extracting both types of information are required, combining them and uncovering their meanings.

To address these challenges, several architectures have been proposed over time: two-stream models that separate spatial from temporal processing [18] [19] [20], 3D CNNs that extend

spatial convolutions into the temporal dimension [21] [22]. An innovative approach is to decouple spatial and temporal modeling into specialized modules - CNNs for spatial features and RNNs for temporal dependencies.

Multimodal models are neural architectures specifically designed to process and integrate information from multiple input modalities, such as visual, text, audio data [23] [24]. These models are essential for tasks requiring complex understanding of the real world, where relying on a single type of signal is insufficient. In the context of driving scenarios, integrating visual data with linguistic inputs and outputs demands such multimodal capabilities [25] [26] [27]. As a result, multimodal learning enables the model to combine visual and linguistic representations, generating outputs based on what the model "sees."

F. Convolutional Neural Networks (CNNs)

For running on edge devices with limited computational resources, compact CNN architectures such as MobileNet [28] [29] and EfficientNet [30] have been developed. Some of the techniques used by these models include depthwise separable convolutions and inverted residuals to reduce computational load while maintaining performance. These lightweight CNNs are necessary to develop a solution capable of running on edge devices.

G. Recurrent Neural Networks (RNNs)

In many real-world scenarios, including driving situations, there are very strong temporal dependencies. For this purpose, specialized architectures exist, such as Recurrent Neural Networks (RNN), which are designed to process sequential data by maintaining a hidden state across time steps [31] [32]. This allows the model to encode the context and dynamics of temporally ordered inputs. Gated Recurrent Units (GRUs) [33] are a type of RNN designed to address vanishing gradient problem in standard RNNs, while being computationally light.

H. Encoder-Decoder Transformers

Encoder-Decoder Transformers are a type of neural architecture designed for sequence-to-sequence tasks, such as machine translation, text summarization or speech processing [34]. Their distinctive advantage lies in the use of self-attention mechanisms, which enable the modeling of long-range dependencies within sequences. Additionally, the encoder-decoder framework allows the model to generate output sequences based on rich, context-aware representations of the input. These characteristics collectively make Encoder-Decoder Transformers well-suited for tasks requiring complex comprehension and structured generation.

III. Proposed Solution

The proposed multimodal architecture processes video input and generates a textual output that conveys both the assessed risk level and the corresponding explanation. It is composed of six core components (see Figure 1):

- Frame Extractor
- Visual encoder (CNN)
- Temporal Encoder (RNN)
- Projection Layer
- Transformer
- Token Decoder

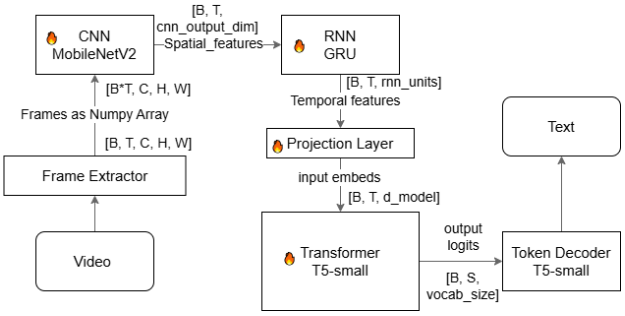

Fig. 1. Model's Architecture

The Frame Extractor component is designed to sample frames from a video file at regular time intervals. The frames are grouped in batches and they get outputted in the following format [B, T, C, H, W], where B is batch size (for multiple videos, when training), T is timesteps (number of frames), C is channels (3 for RGB) and H,W are height and width, which are converted to 224x224.

Before entering the CNN, the frames data is flattened to [B*T, C, H, W] as CNNs expects input of shape size 4 (batch, channels, height, width). By flattening the batch and temporal dimensions the model treats each frame independently during CNN feature extraction and leverages GPU parallelism (the CNN processes all frames in one forward pass, avoiding slow per-frame loops).

The GRU processes CNN features, enhancing them into a higher-dimensional representation that encodes temporal dynamics. Next, the projection layer aligns these features with the transformer's input, which generates the final text output.

The modularity of the architecture enables a clear separation of responsibilities, facilitating training, testing, and adaptation for each individual component. Thus, training can be carried out in multiple stages—starting with the GRU, projection, and T5 decoder [35]—and concluding with the training of all four components in the final stage (see Figure 5). This gradual approach stabilizes the learning process and avoids abrupt overwriting of pretrained parameters. Moreover, this incremental process maximizes knowledge transfer and reduces the risk of overfitting.

The architecture is designed to be computationally efficient, considering the potential integration on edge devices and usage in real-time inference scenarios. The choice of components that prioritize efficiency and minimal memory consumption, balanced with delivered performance, such as MobileNetV2 as the CNN and T5-small [35] as the transformer, reflects optimizations aimed at addressing challenges and meeting constraints: lightweight and deployable. Additionally, the model not only classifies but also generates textual explanations, fulfilling the interpretability constraint.

A. Risk Assessment Dataset Generation

To enable robust risk assessment and provide complex explanations of the driving scenarios, this chapter focuses on generating a structured dataset, that will be used for training Vision-Risk model, containing diverse driving scenarios with annotated risk levels and explanations. The goal is to compile a comprehensive collection that has the structure presented in Figure 2.

file_path	start	end	danger_ label	description
<example_path>	<start_in _ms>	<end_in _ms>	<label>	**Danger level: <label>** <explanation>

Fig. 2. Dataset Structure

- **Generating the Explanations for Low and Medium Risk Scenarios**
  Understanding driver behavior in low and medium risk scenarios is critical for developing safer autonomous systems as they are very common in daily driving. We leveraged Honda Driving Dataset (HDD) [15] which provides a rich collection of real-world driving sequences as it contains cognitive labels (described in Section II). This in combination with Dolphins model and Deepseek V3 [36], allows us to generate human-interpretable explanations that highlight key factors and driver responses.

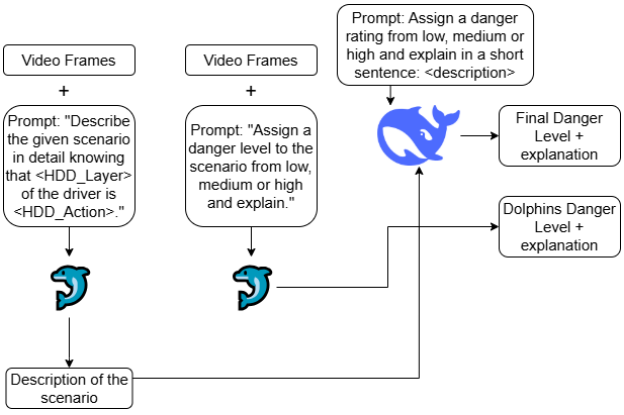

Fig. 3. Strategy for generation of Low/Medium Risk Explanations

We adopted an approach that uses Dolphins to generate explanations for each driving scenario, primarily due to scarcity of explainable data in the domain. This choice was further motivated by Dolphins' ability to produce coherent explanations based on its understanding of video data, enabling a knowledge transfer from a large-scale model to our smaller and lightweight model. The generated explanations serve as synthetic labels, significantly accelerating the creation of a large dataset without needing manual annotation.

As Dolphins excels in providing contextually rich descriptions, we employed DeepSeek V3 due to its advanced reasoning capabilities. This approach demonstrates superior performance in extracting the actual risk factors from Dolphins' contextual descriptions and providing corrected risk classifications while maintaining logical consistency across dataset. Figure 3 illustrates this process, showing how these two models were leveraged to enrich the dataset with textual rationales.
While Dolphins remains valuable for environmental context generation, DeepSeek proved essential for: superior reasoning, risk assessment accuracy and dataset coherence maintenance.

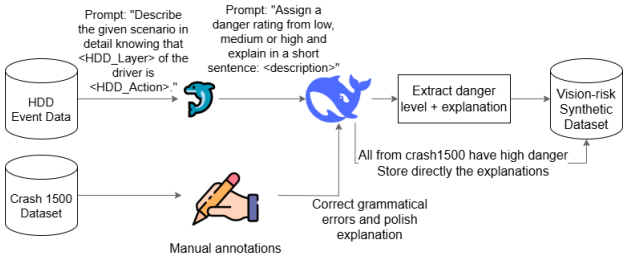

Fig. 4. Pipeline for generating the full dataset

- **Generating the Explanations for High Risk Scenarios**
  In contrast to HDD, which contains numerous sequences of regular daily driving, Crash1500 [37] is a dataset that focuses exclusively on high-risk scenarios. Integrating these events into the VISION-RISK training dataset is essential for accurately modeling dangerous situations and for learning robust representations of risky behaviors. This integration ensures a balanced distribution, enabling the model to learn both normal-risk cases and extreme ones. However, the same issue mentioned earlier arises here as well—the lack of labeled data—highlighting the need to supplement the dataset with manually annotated examples as shown in Figure 4.
  A challenge encountered with Dolphins is its inability to accurately recognize crash and collision scenarios, as the dataset it was trained on did not include such cases. To address this, textual explanations for the videos from Crash1500 were manually created, serving as a reference point for calibrating and validating the manually generated labels. Furthermore, in high-risk situations, automatic labeling cannot be fully relied upon, necessitating human supervision to ensure the accuracy and coherence of the annotations.
  Manually written explanations often contain grammatical errors, redundant expressions, or stylistic inconsistencies that compromise the quality of linguistic supervision. Therefore, to refine these explanations and obtain a clean, professionally written dataset, we employed again DeepSeek. Its purpose is clear: to perform grammatical correction, sentence rewriting, and formulation standardization—without altering the content, meaning, or logic

of the explanations. This step is essential for reducing noise in the training data and enhancing the model's ability to generate coherent and natural explanations.

TABLE I
DANGER LEVEL BREAKDOWN: FULL VS. BALANCED DATASET

Danger Level	Full Dataset (%)	Balanced Dataset (%)
Low	69.3	52.9
Medium	25.8	33.0
High	4.9	14.1

The extended dataset contains 32,469 scenarios with a natural distribution as seen in table I. Due to the scarcity of extreme risk scenarios, it struggles to learn from high-risk data. The model's inability to learn high-risk instances represents a critical issue, and for this reason, in the prototyping phase, we considered that it is better to have a more balanced distribution as seen in table I (about 15000 scenarios). After splitting, in the training dataset, we addressed the class imbalance by duplicating the high-risk scenarios three times, in order to force the model to better learn and generalize these underrepresented cases.

Stage	CNN	GRU + Projection	T5 Encoder	T5 Decoder	Use Case
1	✗	✓	✗	✓	Align temporal –> text
2	✗	✓	✓	✓	Fine-tune full text encoder
3	✓	✓	✓	✓	End-to-end visual learning

Fig. 5. Multi-Stage Training Strategy

B. Multi-Stage Training Strategy

Training deep multimodal architectures end-to-end from the beginning can lead to catastrophic results, such as instability, vanishing gradients or even the model "forgetting" previously learned knowledge—especially when combining multiple pre-trained components [38]. To address these challenges, we adopted a multi-stage training strategy, where modules are gradually introduced into the learning process (see Figure 5). This approach allows the model to first learn stable representations and output formats before fine-tuning the entire architecture in a unified manner.

The loss function that we used is Cross-Entropy, measuring how different is the predicted probability distribution from the true one.

IV. EXPERIMENTAL WORK

In the context of our model, VISION-RISK, which is designed to predict risk levels and generate natural language explanations, it is important to evaluate both outputs: to measure the accuracy and reliability of the model's classifications, and to assess the clarity, relevance, and coherence of the generated explanations. A comprehensive evaluation must incorporate both quantitative and qualitative perspectives to fully capture the model's effectiveness and trustworthiness.

Metrics such as accuracy, precision, recall, and F1-score are essential for assessing the danger level classification task. Similarly, text generation metrics like BLEU [39], ROUGE [40], and BERTScore [41] allow for measuring the lexical and semantic similarity between the model's generated explanations and reference texts.

In the following plots, we present key evaluation metrics and highlight significant changes in both the convergence behavior and the evolution of the model's ability to detect risk levels and produce coherent, context-aware explanations.

- The most significant loss evolution was in stage 1.
- In stage 1 we've also observed that the High classification converges to nearly 1.
- For classification, in stage 1 it looks like the high and low are on rise, while medium it does not get much better.
- In stage 2 and 3 we do not see much improvement in classification, but rather in semantic metrics, where the model gets better at providing textual explanations as seen in Figure 6.
- By Stage 3, metric gains plateaued, and early signs of overfitting to the training data became apparent, with little to no further enhancement in performance (see Figure 7).

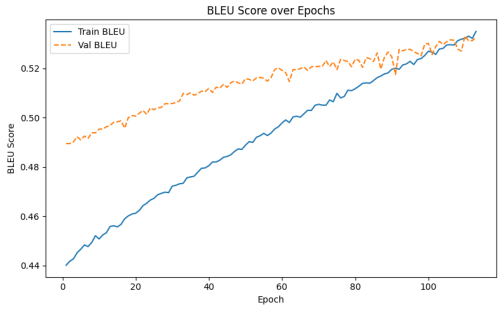

Fig. 6. BLEU Score Stage 2

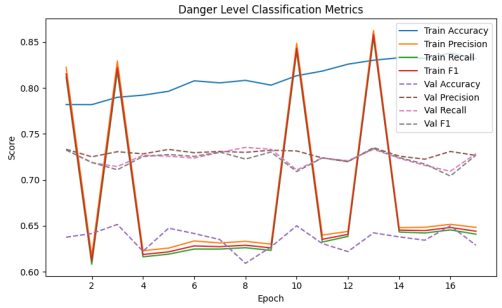

Fig. 7. Stage 3 Danger Classification Metrics

Table II illustrates the evaluation metrics for our best model, that we obtained through training. The results prove that the model can achieve significant performance in assessing risk level and providing a complex textual explanation. The model exhibits nearly perfect precision and recall for high danger scenarios. However, a poor performance can be observed in the

classification of medium risk scenarios. This behavior might be a result of the inherent subjectivity involved in evaluating medium-risk scenarios, where risk factors can be ambiguous. In some cases, it is hard to distinguish between low-risk and medium-risk, which leads to inconsistent interpretations and confusion. In terms of semantic fidelity, the metrics illustrate a very high BERTScore, while BLEU, ROUGE scores demonstrate that the model produces high-quality text, very similar to the reference text. The small difference between train and validation metrics indicate that there is limited overfitting. Overall, the model performs reliably, but further refinement is necessary to enhance its ability to detect medium-risk situations, which can be considered a more ambiguous and subjective class.

However, quantitative metrics alone are insufficient in contexts where interpretability and user trust are critical—such as autonomous driving and safety analysis. Qualitative evaluation complements quantitative results by enabling human-centered judgments on the informativeness, correctness, and contextual alignment of the generated explanations. It helps identify issues such as vague language, hallucinated justifications, or inconsistencies between the predicted risk level and its accompanying explanation [42]. The qualitative metrics showed that the model generates a lot of keywords specific to the domain such as „traffic", „intersection", „driver". Also we have discovered some ambiguity when explaining some high danger cases, where the risk level was assigned correctly, but the explanation was not explaining clearly why the situation was dangerous. This might be due to their very diverse nature, that makes it harder to generalize them. In terms of low risk level, there wasn't much ambiguity and the explanations were straight to the point. There were also some cases where there was a confusion, as there were not many potential dangers, but the model still classified the scenario as medium even though it seemed to not be so risky, but this is subjective.

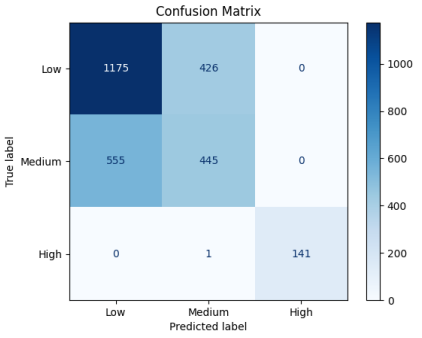

Fig. 8. Confusion Matrix for the Best Model (Stage 3 Epoch 6)

The confusion matrix in Figure 8 reveals that the model performs well on Low-risk scenarios, achieving good precision and recall, though some confusion with the Medium class persists. In contrast, Medium-risk cases show lower precision and recall, as they are frequently misclassified as Low, suggesting the model has difficulty distinguishing this intermediate category. This can be due to the subjectivity in the potential dangers that may be involved. High-risk scenarios are identified with near-perfect accuracy, likely due to targeted oversampling during training, which has enhanced the model's ability to recognize this critical class.

TABLE II
TRAINING AND VALIDATION METRICS FOR BEST MODEL

Metric	Validation	Train
Global Metrics		
Loss	0.1443	0.1355
BLEU	53.2	53.68
ROUGE-1	74.15	75.01
ROUGE-2	53.39	53.92
ROUGE-L	72.98	73.57
BERTScore (Precision)	92.71	92.70
BERTScore (Recall)	93.98	93.96
BERTScore (F1)	93.34	93.32
Danger Level Classification (Overall)		
Accuracy	64.16%	80.78%
Precision	72.96%	84.46%
Recall	72.37%	83.29%
F1-score	72.56%	83.75%
Per-Class Classification Metrics		
Low		
Precision	67.9%	79.61%
Recall	73.32%	85.76%
F1	70.51%	82.58%
Medium		
Precision	50.97%	73.91%
Recall	44.5%	64.95%
F1	47.52%	69.14%
High		
Precision	100%	99.88%
Recall	99.29%	99.18%
F1	99.65%	99.53%

All comparisons reported in Table III were conducted on our held-out test dataset, ensuring that VISION-RISK's superior performance over the raw Dolphins outputs reflects true generalization rather than overfitting to the training data. VISION-RISK outperforms the raw Dolphins outputs across every major metric. These results clearly demonstrate that VISION-RISK not only provides more accurate danger-level predictions but also generates far more coherent and semantically faithful explanations than the Dolphins model in the context of risk assessment.

During manual inspection of Dolphins' generated explanations, we identified instances of semantic hallucinations where the model's predicted risk level contradicted its own textual descriptions. A recurring pattern emerged where Dolphins would accurately describe the environment (e.g., "the car makes a right turn at an intersection") but draw incorrect risk conclusions (labeling it as high risk despite the intersection being empty).

The optimizer updates model weights based on the gradients computed during the backpropagation. In our case, we used AdamW [43], which is a variant of Adam [44] that decouples weight decay from the gradient update, which improves performance for transformer-based models like T5.

A critical aspect of our optimization approach involves the implementation of differential learning rates across distinct

TABLE III
COMPARISON BETWEEN DOLPHINS AND VISION-RISK ON THE TEST SET

Metric	Dolphins	VISION-RISK
Global Metrics		
BLEU	1.4	53.14
ROUGE-1	23.56	74.01
ROUGE-2	3.23	53.18
ROUGE-L	17.38	72.81
BERTScore (Precision)	88.06	92.75
BERTScore (Recall)	86.39	94.00
BERTScore (F1)	87.21	93.37
Danger Level Classification (Overall)		
Accuracy	39.65%	63.48%
Precision	36.13%	72.53%
Recall	43.00%	71.85%
F1-score	35.87%	72.13%
Per-Class Classification Metrics		
Low		
Precision (Low)	59.50%	67.72%
Recall (Low)	38.70%	71.78%
F1 (Low)	46.89%	69.69%
Medium		
Precision (Medium)	32.09%	49.88%
Recall (Medium)	39.6%	45.2%
F1 (Medium)	35.45%	47.42%
High		
Precision (High)	16.82%	100%
Recall (High)	50.70%	98.59%
F1 (High)	25.26%	99.29%

model components as seen in Table IV: the CNN should be fine-tuned slowly with a small LR, the GRU and projection should learn faster since they are randomly initialized and t5 has a small LR to avoid destroying pretrained language generation capabilities.

TABLE IV
LEARNING RATES ASSIGNED TO EACH MODEL COMPONENT IN THE OPTIMIZER CONFIGURATION

Model Component	Learning Rate
CNN	1×10^{-5}
GRU	1×10^{-3}
Projection Layer	1×10^{-3}
T5	2×10^{-5}

A GradScaler was used for Automatic Mixed Precision (AMP) training, as it prevents numerical underflow during backward passes when using float16 precision. It also scales the loss before backpropagation, then unscales gradients before optimizer steps, allowing faster training and reduced memory consumption on GPUs.

The model was trained across three stages using three NVIDIA L40S GPUs, with the complete training process spanning approximately 2.5 days until convergence was achieved. To ensure efficient training while avoiding overfitting, an early stopping criterion was employed: training was halted if the validation loss did not improve for five consecutive epochs. This strategy was found to be effective in identifying the point of convergence without unnecessary computational overhead.

The trained model was deployed on a Raspberry Pi 5 along with a simple Web UI (see Figure 9) to evaluate inference performance in a low-power edge computing environment and

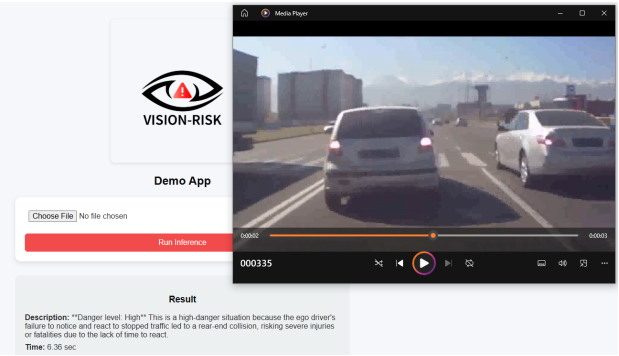

Fig. 9. Inference Example on Raspberry Pi 5

the quality of generated explanations. The average inference time was approximately 6 seconds per sample. This result was obtained without applying any optimization techniques, such as model format conversion (e.g., to ONNX) or graph-level simplifications, and without the use of external hardware accelerators. These findings demonstrate the model's potential for real-time deployment, even in its unoptimized state.

We have evaluated the model in several zero-shot generalization scenarios using various driving videos randomly sourced from the Internet and the key observations are as follows:

- The model identified correctly all high-risk scenarios. However, there were some inaccuracies in the generated explanations (mislabeling an overtaking maneuver as a U-turn or ambiguous phrasing when it did not refer correctly to the vehicle that caused the accident). Nevertheless, the model largely succeeded in identifying key risk factors such as failure to brake, right-of-way violations, inability to react, or lack of situational awareness.
- When testing on low and medium risk scenarios, the model encountered difficulties in recognizing them in different road environments. However, for randomly selected Internet videos filmed within the San Francisco region, the model was able to correctly identify the risk, as the training data for these kinds of scenarios was recorded in that geographical area.

V. CONCLUSION AND FUTURE DIRECTIONS

In the pursuit of safer and more interpretable systems in autonomous driving, we proposed VISION-RISK as a novel multimodal model capable not only of assessing driving risk but also of generating human-readable textual explanations for its predictions. By addressing the dual challenge of accurate danger classification and generation of complex explanations, VISION-RISK represents a step toward more accountable and trustworthy AI systems in the driving domain. To enable this, a semi-synthetic dataset was developed by combining real driving data with synthetic annotations, allowing the model to learn both risk detection and explanatory reasoning. To validate its applicability in real-world edge scenarios, VISION-RISK was deployed on a Raspberry Pi 5, where it

achieved a low inference time without any hardware acceleration or optimizations, demonstrating the model's feasibility for lightweight, low-cost deployment in constrained environments.

These findings lay the groundwork for future research aimed at enhancing both the efficiency and capability of VISION-RISK. Promising directions include the application of model optimization techniques—such as quantization, pruning, or ONNX conversion—to significantly reduce inference time, alongside the integration of lightweight hardware accelerators to support real-time deployment on embedded systems. Moreover, expanding the dataset with increasingly complex and diverse driving scenarios, as well as incorporating additional sensory modalities such as LiDAR and radar, holds the potential to further strengthen the model's robustness and explanatory depth.

ACKNOWLEDGMENT

This manuscript is a non-peer-reviewed version of the research performed by the listed authors. A later, revised version of this work was submitted to and accepted by the ICCP 2025[1] conference, but the submission and final author list were made without the knowledge or consent of Sebastian-Antonio Toma. This version is being posted to document the intellectual contributions of all original authors.

REFERENCES

[1] D. Coelho and M. Oliveira, "A review of end-to-end autonomous driving in urban environments," *IEEE Access*, vol. 10, pp. 75296–75311, 2022.

[2] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, "End-to-end autonomous driving: Challenges and frontiers," 2024.

[3] D. Fu, X. Li, L. Wen, M. Dou, P. Cai, B. Shi, and Y. Qiao, "Drive like a human: Rethinking autonomous driving with large language models," 2023.

[4] J. Mao, Y. Qian, J. Ye, H. Zhao, and Y. Wang, "Gpt-driver: Learning to drive with gpt," 2023.

[5] Y. Jin, R. Yang, Z. Yi, X. Shen, H. Peng, X. Liu, J. Qin, J. Li, J. Xie, P. Gao, G. Zhou, and J. Gong, "Surrealdriver: Designing llm-powered generative driver agent framework based on human drivers' driving-thinking data," 2024.

[6] C. Sima, K. Renz, K. Chitta, L. Chen, H. Zhang, C. Xie, J. Beißwenger, P. Luo, A. Geiger, and H. Li, "Drivelm: Driving with graph visual question answering," 2025.

[7] Z. Xu, Y. Zhang, E. Xie, Z. Zhao, Y. Guo, K.-Y. K. Wong, Z. Li, and H. Zhao, "Drivegpt4: Interpretable end-to-end autonomous driving via large language model," 2024.

[8] Wayve, "Lingo-1: Exploring natural language for autonomous driving." https://wayve.ai/thinking/lingo-natural-language-autonomous-driving/.

[9] L. Petersen, H. Zhao, D. Tilbury, X. J. Yang, and L. Robert, "The influence of risk on driver trust in autonomous driving systems," 11 2024.

[10] Y. Ma, Y. Cao, J. Sun, M. Pavone, and C. Xiao, "Dolphins: Multimodal language model for driving," 2023.

[11] A. Awadalla, I. Gao, J. Gardner, J. Hessel, Y. Hanafy, W. Zhu, K. Marathe, Y. Bitton, S. Gadre, S. Sagawa, J. Jitsev, S. Kornblith, P. W. Koh, G. Ilharco, M. Wortsman, and L. Schmidt, "Openflamingo: An open-source framework for training large autoregressive vision-language models," 2023.

[12] X. Yu, C. Zhou, Z. Kuai, X. Zhang, and Y. Fang, "Gcot: Chain-of-thought prompt learning for graphs," 2025.

[13] OpenAI, "Chatgpt: Language models for dialogue." https://openai.com/chatgpt.

[14] J. Kim, A. Rohrbach, T. Darrell, J. Canny, and Z. Akata, "Textual explanations for self-driving vehicles," *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[1]https://www.iccp.ro/technical-program

[15] V. Ramanishka, Y.-T. Chen, T. Misu, and K. Saenko, "Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning," 2018.

[16] D. Xiao, M. Dianati, P. Jennings, and R. Woodman, "Hazardvlm: A video language model for real-time hazard description in automated driving systems," *IEEE Transactions on Intelligent Vehicles*, pp. 1–13, 2024.

[17] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," 2020.

[18] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," 2014.

[19] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," 2016.

[20] H. Ye, Z. Wu, R.-W. Zhao, X. Wang, Y.-G. Jiang, and X. Xue, "Evaluating two-stream cnn for video classification," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, ICMR '15, p. 435–442, ACM, June 2015.

[21] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," 2015.

[22] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," 2018.

[23] M. Suzuki and Y. Matsuo, "A survey of multimodal deep generative models," *Advanced Robotics*, vol. 36, p. 261–278, Feb. 2022.

[24] P. Xu, X. Zhu, and D. A. Clifton, "Multimodal learning with transformers: A survey," 2023.

[25] A. Ishaq, J. Lahoud, F. S. Khan, S. Khan, H. Cholakkal, and R. M. Anwer, "Tracking meets large multimodal models for driving scenario understanding," 2025.

[26] Z. Qiao, H. Li, Z. Cao, and H. X. Liu, "Lightemma: Lightweight end-to-end multimodal model for autonomous driving," 2025.

[27] S. Sreeram, T.-H. Wang, A. Maalouf, G. Rosman, S. Karaman, and D. Rus, "Probing multimodal llms as world models for driving," 2024.

[28] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017.

[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," 2019.

[30] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2020.

[31] R. M. Schmidt, "Recurrent neural networks (rnns): A gentle introduction and overview," 2019.

[32] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks," 2018.

[33] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[35] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2023.

[36] DeepSeek, "Deepseek-v3: Advanced language model." https://deepseek.com.

[37] W. Bao, Q. Yu, and Y. Kong, "Uncertainty-based traffic accident anticipation with spatio-temporal relational learning," in *ACM Multimedia Conference*, May 2020.

[38] D. Zhu, Z. Sun, Z. Li, T. Shen, K. Yan, S. Ding, K. Kuang, and C. Wu, "Model tailor: Mitigating catastrophic forgetting in multi-modal large language models," 2024.

[39] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (P. Isabelle, E. Charniak, and D. Lin, eds.), (Philadelphia, Pennsylvania, USA), pp. 311–318, Association for Computational Linguistics, July 2002.

[40] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, (Barcelona, Spain), pp. 74–81, Association for Computational Linguistics, July 2004.

[41] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," 2020.

[42] I. Ni'mah, M. Fang, V. Menkovski, and M. Pechenizkiy, "Nlg evaluation metrics beyond correlation analysis: An empirical metric preference checklist," 2023.

[43] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” 2019.
[44] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017.