Enhanced Defect Detection in Airport Runway Infrastructure Using Image-Text Pairing

Marios Krestenitis*
Centre for Research and Technology-Hellas
Thessaloniki, Greece
mikrestenitis@iti.gr

Eftichia Badeka
Centre for Research and Technology-Hellas
Thessaloniki, Greece
efibad@iti.gr

Ilias Koulalis
Centre for Research and Technology-Hellas
Thessaloniki, Greece
iliask@iti.gr

Konstantinos Ioannidis
Centre for Research and Technology-Hellas
Thessaloniki, Greece
kioannid@iti.gr

Stefanos Vrochidis
Centre for Research and Technology-Hellas
Thessaloniki, Greece
stefanos@iti.gr

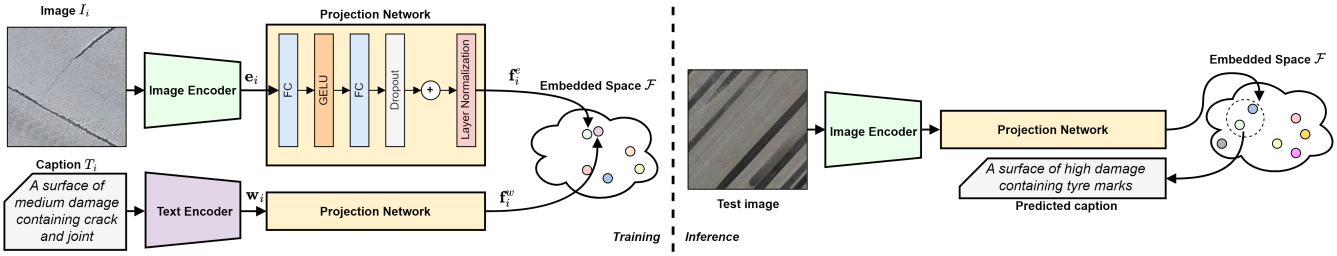

Fig. 1: Overview of the proposed method for CLIP-based defect detection. During training (left) the model learns to associate images of runway surface with text inputs that describe the condition of the surface. In the inference phase (right), the model can process a surface image and provide a text description regarding the existing defects and the damage severity.

***Abstract*—Maintaining runway infrastructure is vital for air transport safety, with defects like cracks and tire marks posing significant risks to the take-off/landing process. Researchers have proposed various methods for automatic detection of surface defects, using computer vision and machine learning. However, they often require explicitly annotated datasets that demand significant workload and field expertise. Additionally, the detection outcome usually follows the low-level training labels scheme to describe the detected defects, and requires post-processing for high-level semantic information extraction, such as damage level estimation. In this work, we present a novel method for defect detection and damage severity estimation on runway surfaces, leveraging the Contrastive Language-Image Pre-training (CLIP) architecture for image-text pairing. Our model processes runway images and attaches text descriptions mentioning detected defects and severity level, identifying three defect types (crack, joint, and tire mark) and categorizing damage severity into three levels (low, medium, and high). Utilizing natural language annotations simplifies the labeling process, eliminating the need for labor-intensive low-level image-based annotations. The model exploits the high-level natural language annotation for direct estimation of damage severity and delivers high-level semantic information to the end-user as text, providing a comprehensive runway condition assessment tool. The proposed method demonstrates high performance across various test sets, posing a valuable human-centric approach for efficient defect detection and damage estimation on runway surfaces.**

***Index Terms*—Defect Detection, Construction Monitoring, CLIP, Human-centric Approaches**

I. Introduction

Airports serve as critical transportation hubs, facilitating the movement of passengers and goods across the globe. Detecting and addressing defects in airport infrastructure is essential for maintaining operational continuity and mitigating potential safety hazards [1]. Defects such as cracks, potholes, surface irregularities like tire marks, and structural damage, can compromise the integrity of runway, leading to operational disruptions and safety risks during takeoff and landing procedures. Traditional methods for defect detection and operational maintenance often rely on visual on-the-spot inspections, scheduled multiple times daily. However, besides being time-consuming and potentially underestimating damage, this approach also exposes airport personnel to risks by requiring them to work directly on the runway.

The integration of computer vision techniques into defect detection processes offers several advantages over traditional manual inspection methods, enabling more rapid and accurate defect identification across large volumes of products [2]. Integrating these capabilities with smart sensory systems, like UAVs, airports can autonomously conduct defect detection surveys with efficiency and precision [3], [4]. This reduces reliance on manual labor while enhancing the overall safety and reliability of airport operations. The typical computer vision-based approach includes analyzing the captured imagery and automatically detecting the depicted defects. To this end, several image processing methods have been employed to extract and identify defect-related features from images. The advancements in Deep Learning (DL) and Convolutional Neural Networks (CNNs) have been employed in this domain also, to provide more sophisticated solutions. In this light, remarkable DL-based methods have been introduced to detect various types of defects, even under challenging conditions

*Corresponding Author

such as variations in size/shape, surface textures, etc. These methods are capable of producing detailed defect detections, sometimes even at the pixel level, enabling accurate identification across a range of anomalies.

Nevertheless, these DL-based methods often rely on extensive annotated datasets for model training and evaluation. However, the annotation process is inherently tedious and time-consuming, requiring meticulous attention to detail, consistency and domain expertise. Moreover, the annotated data must adhere to specific model-oriented schemes, such as pixel coordinates of bounding boxes, pixel-wise masks, or numbered labels for different classes. Even when the decision-making process of labeling instances is straightforward, adhering to specific annotation schemes can still prove to be tedious and time-consuming for annotators. Furthermore, this type of image-level annotations typically provide low-level information, vaguely describing defects' location and shape. Hence, prediction outcome also follows this low-level scheme, describing the detected defects in a low-level concept that is not directly plausible from the end-user. Moreover, it usually requires further post-processing of the detection result to estimate meaningful information about the examined infrastructure, e.g. quality of the damaged surface.

Employing a higher-level annotation scheme, which is human-centric instead of tailored to the model design, such as using natural language to label images, can enhance the defect detection process in several ways. First, it simplifies the annotation process by describing the depicted semantic content (e.g. "a wall surface with cracks"), while reducing the need for field experts. Natural language allows for describing high-level concepts depicted in the scene (e.g. "an outside wall surface with several cracks and spalling that needs some repair"). Contrary to low-level image-based labels, utilizing this high-level information guides a model to learn high-level concepts that are enclosed in the training data and thus, associate the text-described surface defects and the wall condition with the visual attributes of the input images. Hence, a model capable to pair images to text descriptions can exploit these attributes and provide high-level semantic information in the form of text captions, that are directly accessed from the end-user and describe high-level concepts regarding the infrastructure, e.g. "a damaged concrete surface with cracks and peeling paint that needs to be repaired".

In this light, we present a novel vision-language framework for defect detection and damage estimation on runway surfaces. The proposed method is built upon the well-known Contrastive Language–Image Pre-training (CLIP) [5] model, a deep-learning architecture capable of pairing images to text. As illustrated in Figure 1, during training, we attach text descriptions to runway images, specifying the depicted defects and the estimated damage level. The designed model is trained to associate this natural language description with the visual concept of the image. In inference mode, our framework encodes a given runway image and attaches to it the text caption whose feature vector best matches the encoded representation, mentioning the detected classes and the estimated severity of damage. The proposed model detects three types of defects - cracks, joints (repaired cracks) and tire marks - and categorizes the surface damage in three levels: low, medium and high. Leveraging natural language simplifies the annotation process and allows providing detection results that are directly accessed from the end user. Furthermore, the model exploits the high-level information from text descriptions to enhance its robustness and efficiently learn the visual semantics related to defects and damage severity — an aspect almost inaccessible through traditional image-based annotation. Overall, we consider our framework a novel human-centric approach for defect detection and damage estimation, comprising a comprehensive tool for runway condition assessment.

II. Related Work

Research community of computer vision and machine learning has proposed several methods for defect detection on construction infrastructures [6], [7]. Early approaches employed a variety of well-known image processing methods to distinguish the depicted defect instance from the background surface. To this end, methods utilizing techniques such as image filtering [8], image thresholding [9], edge detection [10] and image enhancement [11] have been proposed, aiming to detect surface defects (usually cracks).

Advancements in deep learning have also been leveraged in the domain of defect detection. In this light, a set of works have been proposed based on deep learning architectures to detect defects on images of infrastructure surfaces. Authors in [12] developed a decision-making tool for buildings inspection by fine-tuning a Deep Convolutional Neural Network (DCNN) for surface crack detection. A comprehensive CNN-based framework was developed in [13], capable to detect structural cracks and estimate their real-world location. Zhang et al. [14] built a custom dataset to develop a CNN model for defect detection in pavement imagery. A CNN-based model, named CrackNet, was introduced in [15] to identify cracks at pixel level by preserving the spatial dimensions of the input image. Similarly, authors in [16] proposed DeepCrack, a DL network to semantically segment crack instances from the background. The semantic segmentation approach was also employed in [17] to identify crack defects on road surfaces. Authors in [18] proposed a DCNN-based method that assigns bounding boxes on surface defects over airport's runway, while Makekloo et al. [19] explored the utilization of dashcam imagery for the detection of runway distresses.

Although the employment of natural language in defect detection remains to some extent unexplored, a set of works have been proposed in this direction. Authors in [20] proposed WinCLIP, a method based on a set of multi-scale CLIP encoders for defect detection on objects. Zhou et al. [21] proposed AnomalyCLIP, a method utilizing object-agnostic text prompts for defect/anomaly detection across various domains. A CLIP-based architecture was employed in [22] for defect detection on wall surfaces and combined with prompt engineering to optimize model's performance. Similarly, Cao et al. [23] developed a framework for zero-shot defect segmentation

by utilizing field-expert text prompts and target image-based prompts.

Inspired by prior works, our method relies on a CLIP-based network capable of detecting various types of defects and assessing the severity of damage across the surface. Moving a step forward from the previous approaches, we implement our method in a realistic scenario, employing real-world data instead of simplified datasets that streamline the problem (e.g. featuring only one type of defect per image, objects depicted from consistent distance/angle, uniform background texture, constant lighting conditions, etc.). Furthermore, our approach harnesses the power of CLIP technology not only to detect defects but also to provide insightful captions that categorize the severity of detected damage and thus, provide valuable contextual information, facilitating efficient maintenance and safety measures for runway infrastructures.

III. Methodology

A. CLIP Architecture

The core element of the proposed framework is a CLIP-based network. CLIP is a novel architecture designed to associate visual concepts with natural language. In specific, CLIP is capable to estimate how well a given image and a text description fit together. To this end, the CLIP architecture is based on an image and a text encoder to initially encode image and text inputs to vectors, respectively. The encoded representations are projected to a multi-modal embedded space, through a trainable mapping network, where they can be compared to each other. During training, a set of image-text pairs is fed to the network and it is estimated the similarity of each image to every text prompt in the embedded space, in terms of cosine similarity. The overall network is trained with a contrastive objective function that aims to maximize the similarity of the correct image-text pair and minimize the similarity with rest incorrect pairs. In the inference mode, the trained model can be fed with a single image and a set of candidate text descriptions and estimate which description fits the best to the provided image. Towards this direction, one can easily assign text to images, that describe the visual concept or even classify the images according to it. The specific design of CLIP enables zero-shot performance, since there is no restriction for specific labels in the inference phase.

B. Proposed Method

The proposed method follows the aforementioned scheme in order to attach text descriptions to runway surface images. In specific, a CLIP-based network is designed to recognize possibly existing defects and characterize the condition of the depicted runway surface, in terms of damage severity. An overview of the proposed method is presented in Figure 1. Our framework is composed of three main elements, namely the image encoder, the text encoder and the projection network.

During training each image $I_i \in \mathbb{R}^{m \times n \times 3}$ of the training set and its corresponding text description T_i are fed to the image and text encoder, respectively. The image encoder process I_i and leads to a compact vector representation $\mathbf{e}_i \in \mathbb{R}^N$, that encloses high-level visual features of the input imagery. Correspondingly, the text encoder process the text input T_i and leads to a vector $\mathbf{w}_i \in \mathbb{R}^M$ that encodes the provided natural language context. For the image encoder the ResNet [24] deep-learning architecture is employed, specifically ResNet50, which is well-known for its robustness. While the text encoder is a DistilBERT [25] model, a Transformer network based on BERT [26] architecture (yet significantly lighter) for efficient language representation. Each one of the two vectors $\mathbf{e}_i$ and $\mathbf{w}_i$, acquired from the image and text encoder, respectively, enclose a different modality of the input information. By comparing $\mathbf{e}_i$ and $\mathbf{w}_i$ one can estimate how closely the two modalities describe the common semantic content of the input. Yet, since $\mathbf{e}_i$ and $\mathbf{w}_i$ differ in size, they remain incomparable to each other. Thus, they should be transferred in common multi-modal feature space where the similarity estimation among the different modalities is enabled.

To achieve this, a projection network is utilized to map each encoded vector to the common K-dimensional embedded space $\mathcal{F}$. As illustrated in Figure 1, the encoded vector $\mathbf{e}_i \in \mathbb{R}^N$ is forwarded to the projection network, which adopts a simple architecture consisting of fully-connected layers. Specifically, $\mathbf{e}_i$ is fed to a fully connected layer to transform it to a K-sized vector, aligning with the dimensionality of the embedded space $\mathcal{F}$. Next, the GELU [27] activation function is applied to the output of the fully connected layer, followed by another fully connected layer with dropout. Similarly to a ResNet-like architecture, the output of the first fully connected layer is added to the output of the second fully connected layer. Lastly, layer normalization is applied, yielding the projected vector $\mathbf{f}_i^e \in \mathbb{R}^K$ of the embedded space $\mathcal{F}$. Similarly, the output of the text encoder $\mathbf{w} \in \mathbb{R}^M$ is processed by another projection network, sharing the same aforementioned architecture, resulting in the transformed vector $\mathbf{f}_i^w \in \mathbb{R}^K$ within $\mathcal{F}$.

Since the two encoded vectors have been mapped to the common space $\mathcal{F}$, the similarity among the image-based and the text-based embedded vectors $\mathbf{f}_i^w$ and $\mathbf{f}_i^w$, respectively, can be easily estimated by calculating their dot product. During training, the dot product among each image and text embedded vector of the batch is calculated, and every image is paired with the text that leads to the maximum dot product value. Thus, the objective function is to minimize the cross-entropy loss among the estimated pairs and the correct ones.

As depicted in Figure 1, in inference mode the trained framework is utilized to attach a text description to a previously unseen image. To this end, the provided image is fed to the model and projected to the embedded space $\mathcal{F}$, where the dot-product similarity among a set of candidates embedded text descriptions is measured. Finally, the text description with the most similar embedded vector is selected as the caption of the given image.

The deployed framework can significantly advance the process of defect detection over runway areas. By training the model with paired images and texts that describe the condition of the depicted surface - naming the captured defects

and the severity of the damage - it can learn to associate the given visual content with the high-level semantic information of the text caption. Unlike traditional image-based detection methods, which typically rely on labor-intensive, low-level annotations tailored to the model architecture (e.g. bounding box coordinates, pixel-wise annotations masks, class labels, etc.), our approach enables the use of simple natural language as a high-level annotation scheme. This simplifies the labelling process and relaxing the constraint of involving only field experts. Moreover, typical image-based detection methods guided by the aforementioned low-level types of ground truth during training primarily learn to recognize basic visual attributes (e.g. color, shape, geometry, etc.) of the defects. On the contrary, our approach leverages the high-level semantic information of the natural language annotations to recognize high-level concepts in the visual input and thus, detect different types of defects as well as estimate the level of existing damage.

C. Dataset Creation

To deploy the proposed method, we developed an image-text dataset designed for runway monitoring tasks. To this end, we utilized an existing UAV-based dataset, developed for semantic segmentation tasks under the scope of H2020 ASHVIN project [28]–[30]. More specifically, the dataset contains a set of high-resolution UAV images captured over Zadar airport, Croatia. Data were collected via two flights, conducted at different dates and covering different sub-regions of the airport's runway area. During the first mission, 108 images of size 5000×5000 pixels were collected. While from the second mission, 201 images of size 2000×1500 pixels were acquired. The collected images capture a wide variety of textures of the runway's surface. Moreover, field-experts have annotated in pixel-level the collected images, highlighting three main types of surface defects, cracks, joints (repaired cracks) and tire marks.

These UAV images were utilized to create a bigger set of runway surface images, while the corresponding pixel-level annotations to generate the text descriptions, that characterize the condition of the depicted area. In this direction, each image was split to non-overlapping patches of size 112×112, which is half of the typical 225×225 patch size that was used in the original CLIP paper. For every image patch, the corresponding patch from the annotation mask was also cropped. Then, the corresponding text description of the image patch was automatically generated based on its semantic content, which was available through the cropped annotated patch.

Specifically, the generated text description characterize the condition of the depicted surface in terms of existing defects and the severity of damage. Each description follows the format: 'A surface of *[damage level]* damage containing *[detected defects]*.' Where, *[damage level]* represents the severity of damage, categorized as *low*, *medium*, or *high*, while *[detected defects]* encompasses the various types of identified defects. The types of defects within each patch are directly accessible from the corresponding annotation. While, damage severity is estimated by initially calculating the damage percentage d, as described by the following equation:

$$d = \frac{N^{crack} + N^{joint} + N^{tire\ mark}}{W \times H} \tag{1}$$

where N^{crack}, N^{joint} and $N^{tire\ mark}$ represent the number of pixels annotated as crack, joint and tire mark, respectively. W and H denote the width and height of the patch, respectively. Please note that the value of d belongs to the range $[0, 1]$.

Next, the damage level is classified according to the following thresholding:

$$\textit{damage level} = \begin{cases} \text{low} & \text{if } 0 < d < 0.1 \\ \text{medium} & \text{if } 0.1 \leq d < 0.2 \\ \text{high} & \text{otherwise} \end{cases} \tag{2}$$

The specific threshold values were selected by analyzing the created dataset and estimating the d value over the whole set. The aim was to create three main categories of damage level, each of which contains a sufficient number of samples.

Following the aforementioned process, various text descriptions are generated based on the surface condition and depicted defects. Additionally, in case that no defects are present in a patch, the accompanying text description is formed as 'A runway surface without defects'. Similarly, patches depicting regions outside the designated runway area are characterized by the phrase 'An image outside the runway area'. This differentiation serves to distinguish between cases where the depicted runway surface is defect-free and those where the depicted region falls outside the area of interest. By including this characterization, the model is guided to learn the visual features of a defect-free runway surface more effectively. An overview of the aforementioned data-creation process is presented in Figure 2.

Through this approach we developed a new image-text dataset tailored for tasks related to runway monitoring. A total of $47,192$ and $44,421$ image-text pairs were generated, respectively, by processing data from the first and second UAV mission, respectively. As a result, a comprehensive image-text dataset containing $91,613$ (approximately $100k$) samples was created, capable for efficiently training and evaluating the designed CLIP-based network.

IV. Experimental Results

A. Implementation Details

The developed dataset was divided into train and test sets according to the following strategy. Image-text pairs generated from the first UAV mission ($47,192$ samples) were reserved for training only. Meanwhile, image-text pairs derived from the second UAV mission ($44,421$ samples) were randomly split into training (80%) and testing (20%) subsets. Consequently, the training set comprises $82,728$ samples, while the testing set consists of $8,884$ samples. Please note that only image-text pairs from the second UAV mission are employed in testing phase. Instead of a more conventional method of dividing to train-test sets the samples from both UAV missions, we

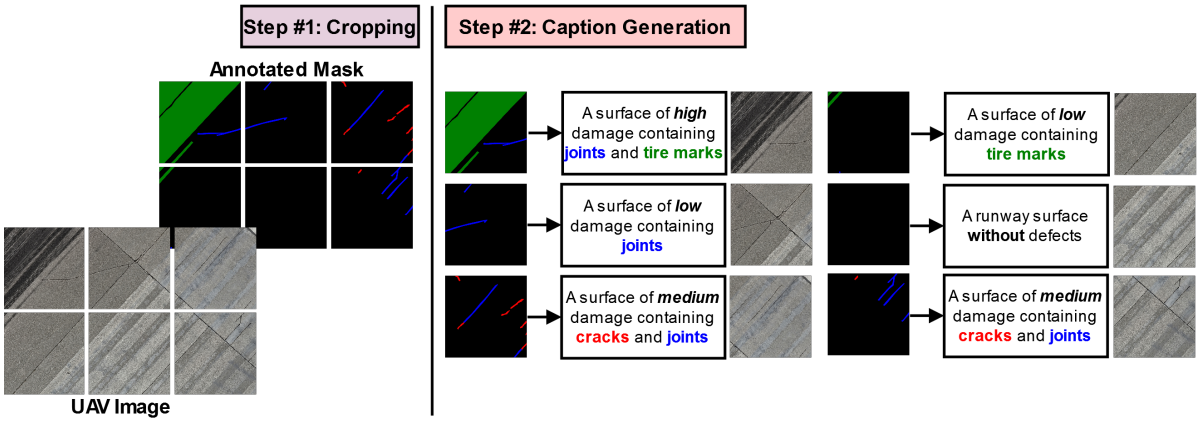

Fig. 2: Dataset creation overview. Non-overlapping patches are cropped from each image and its corresponding annotation mask. Text descriptions are automatically generated for each patch based on its annotation. Cracks are marked in red, joints in blue, tire marks in green, and background in black.

selected this method as a more challenging, yet realistic, approach, since it increases the diversity between the training and testing data. Please note that the non-overlapping approach described in Section III-C guarantees that no information of the training set is leaked to the testing phase.

During training a data augmentation strategy was followed to increase the generalization ability of the model. In particular, every input image was resized to 224×224, which is the go-to input size in the original CLIP work. Additionally, transformations were applied to each input image with a possibility of 0.5 each, namely horizontal flip, vertical flip and rotation within the range $[-10°, 10°]$. The deployed network was trained for 30 epochs with AdamW optimizer and weight decay equal to 0.2 epochs. The learning rate was set equal to $1e-3$, $1e-4$ and 10^{-5} for the projection network, the image encoder and the text encoder respectively. The image encoder leads to embedded vectors of size 2048, while the text encoder to size 768. The projection network maps the vectors of each encoder to the common multi-modal space $\mathcal{F}$ of 1024 dimensions. The image encoder was pretrained in ImageNet [31] while text encoder to BookCorpus [32] dataset. The specific datasets are established for guaranteeing the robustness of the encoders to extract high-quality features and operate efficiently, especially when processing in-domain data. Towards this direction, we kept the weights of the two encoders frozen during the first epoch to avoid backpropagating noisy updates due to the random weight initialization of the projection network and conduct a smooth training process. For the rest of the training epochs, the weights of all submodules are updated to adapt to the special characteristics of the runway dataset, which is considered as out-of-domain, in respect to the datasets that the two encoders were pretrained.

B. Quantitative Results

We assess the performance of the proposed method by evaluating the top-k score across the test set. For each image in the test set, we feed it into the network and estimate the probability of it matching with each possible text description, i.e. different combinations of damage level and depicted defects, defect-less cases, or out-of-interest areas. To ensure a robust evaluation, we adopted the well-known k-fold validation approach. Thus, the random splitting of the created dataset to train-test sets, as described in III-C, was repeated 3 times leading to 3 distinct dataset folds. For each k-fold we keep the model weights of the epoch that led to the maximum top-k score. Table I summarizes the results for top-1 (similar to accuracy metric), top-3 and top-5 scores. Since the created dataset is to some extent imbalanced due to the nature of the problem - images depicting defect-free surfaces are more than those enclosing defects - we also measured the top-1 score, specifically for cases where defects are present. The results of this scenario are presented in the last column of Table I.

TABLE I: Top-k Score (%) Results of the Proposed Method Over the Defect Detection Dataset

Split	Top-1	Top-5	Top-3	Top-1 (Defects Only)
1	86.90	96.73	98.84	71.70
2	87.46	97.52	99.17	71.18
3	87.37	97.39	98.95	72.30
Mean	87.25	97.22	98.99	71.73

Results of Table I imply the robustness of the proposed method. Notably, the model achieves consistently high top-k scores across all three splits, demonstrating high performance. These scores indicate the ability of the model to associate input images to text prompts that accurately describe the semantic content of the scene, i.e. mention existing defects and estimate the severity of the surface damage. Apart from the typical top-k metrics, this is also underscored from the reported top-1 metric for defects only, where the model showcases mean accuracy equal to 71.73%. Thus, the proposed method is a valuable approach to efficiently detect different types of surface defects while simultaneously estimating the quality of runway surface.

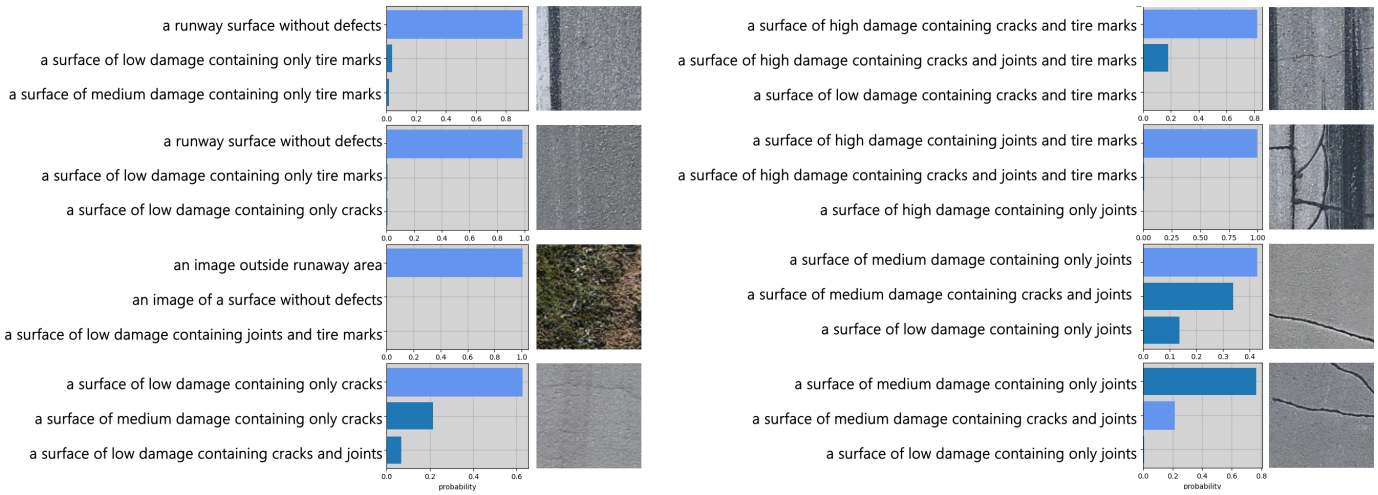

Fig. 3: Qualitative results of the proposed method for a set of test images. For each input image, the top-3 predicted text descriptions are presented among with their corresponding probabilities. The prediction corresponding to the ground truth is highlighted with light blue color.

C. Qualitative Results

Figure 3 presents a set of qualitative results for the proposed method. For each sampled test image, the probabilities of the top-3 predicted text descriptions are reported. Results imply that in the majority of the cases, the developed method can accurately detect the depicted classes and characterize the damage level. It is also interesting to examine the rest two most probable text descriptions predicted by the model. One can notice that the second and third most probable captions are semantically related to the first one. For instance, when the surface has no defects, the second most probable text description is a caption mentioning low damage level. Similarly, when the image depicts an area outside the runway, the second most relevant option is a description of a defect-less surface. In cases where defects are detected, the remaining most probable estimations are also relevant, either in terms of damage level or defect classes. The above indicate that the developed model has effectively learned the semantic content of the images and how the visual information is related to the examined defects and the different severity levels.

V. Conclusions

Monitoring and maintaining airport runway infrastructure is crucial for ensuring high safety standards in air transport. Defects on the runway surface, such as cracks or tire marks, pose significant risks to the take-off and landing process. Researchers haver proposed several approaches to automatically detect these defects, via computer vision and machine learning methods. Nevertheless, a key prerequisite of these approaches is the utilization of relevant explicitly annotated datasets, which demands a heavy workload and field experts. Moreover, the corresponding detection outcome usually follows the low-level scheme of the training labels and requires further post-processing to extract high-level semantic information.

In this work, we present a novel method for defect detection and damage severity estimation on the runway surface based on CLIP architecture. Our model processes runway images and attaches text descriptions indicating defect types (crack, joint, tire mark) and severity levels (low, medium, high), providing a comprehensive assessment of the runway condition. To develop our method, we created a custom image-text dataset, utilizing an existing dataset for semantic segmentation of runway defects and generating text descriptions based on the semantic content. The proposed method presented high performance, in terms of top-k score, across all random three splits of train-test sets. Results imply that the proposed model is a valuable approach that can efficiently detect different types of defects and estimate the damage level on the runway surface. This high-level information is provided in the form of natural language and can be directly accessed by the end-user, enhancing the human-centric design of the system. Moreover, since the annotation process is based on text captions, there is no requirement for specific low-level image-based labels (bounding boxes, pixel-wise masks, etc.) that typically demand extensive labor from field experts.

In the future, we aim to extend our work with additional cross-over experiments and by employing prompt engineering to enrich text captions' semantic information and enhance the model's capabilities. We will also investigate advanced methods for runway condition characterization, e.g. relative construction and material indexes, to shape the text descriptions. Lastly, we will explore using the encoded representations, which contain vital surface condition information, in tasks such as image generation and semantic segmentation.

Acknowledgment

This work was supported by SEISMEC and ASHVIN projects funded by the European Commission under grant agreements No 101135884 and No 958161, respectively.

REFERENCES

[1] J. Rakas, A. Bauranov, and B. Messika, "Failures of critical systems at airports: Impact on aircraft operations and safety," *Safety Science*, vol. 110, no. 3, pp. 141–157, 2018.

[2] J. Landgraf, M. Kompenhans, T. Christ, T. Roland, and A. Heinig, "Computer vision for industrial defect detection," vol. 25. Association of American Publishers, 2023, pp. 371–378.

[3] L. Basora, P. Bry, X. Olive, and F. Freeman, "Aircraft fleet health monitoring with anomaly detection techniques," *Aerospace*, vol. 8, no. 4, 2021. [Online]. Available: https://www.mdpi.com/2226-4310/8/4/103

[4] A. Al-Kaff, D. Martín, F. García, A. de la Escalera, and J. María Armingol, "Survey of computer vision algorithms and applications for unmanned aerial vehicles," *Expert Systems with Applications*, vol. 92, pp. 447–463, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417417306395

[5] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[6] Y. Zhou, X. Guo, F. Hou, and J. Wu, "Review of intelligent road defects detection technology," *Sustainability*, vol. 14, no. 10, 2022. [Online]. Available: https://www.mdpi.com/2071-1050/14/10/6306

[7] W. Cao, Q. Liu, and Z. He, "Review of pavement defect detection methods," *IEEE Access*, vol. 8, pp. 14 531–14 544, 2020.

[8] M. Salman, S. Mathavan, K. Kamal, and M. Rahman, "Pavement crack detection using the gabor filter," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013, pp. 2039–2044.

[9] X.-c. Yuan, L.-s. Wu, and Q. Peng, "An improved otsu method using the weighted object variance for defect detection," *Applied surface science*, vol. 349, pp. 472–484, 2015.

[10] C. Yeum and S. Dyke, "Vision-based automated crack detection for bridge inspection," *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, pp. 759–770, 2015.

[11] B. Lei, N. Wang, P. Xu, and G. Song, "New crack detection method for bridge inspection using uav incorporating image processing," *Journal of Aerospace Engineering*, vol. 31, no. 5, p. 04018058, 2018.

[12] F. Kucuksubasi and A. Sorguc, "Transfer learning-based crack detection by autonomous uavs," *CoRR*, vol. abs/1807.11785, 2018. [Online]. Available: http://arxiv.org/abs/1807.11785

[13] D. Choi, W. Bell, D. Kim, and J. Kim, "Uav-driven structural crack detection and location determination using convolutional neural networks," *Sensors*, vol. 21, no. 8, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/8/2650

[14] C. Zhang, E. Nateghinia, L. Miranda-Moreno, and L. Sun, "Pavement distress detection using convolutional neural network (cnn): A case study in montreal, canada," *International Journal of Transportation Science and Technology*, vol. 11, pp. 298–309, 2022.

[15] L. Zhang, F. Yang, Y. Zhang, and Y. Zhu, "Road crack detection using deep convolutional neural network," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3708–3712.

[16] Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li, "Deepcrack: A deep hierarchical feature learning architecture for crack segmentation," *Neurocomputing*, vol. 338, pp. 139–153, 2019.

[17] S. Bang, S. Park, H. Kim, and H. Kim, "Encoder–decoder network for pixel-level road crack detection in black-box images," *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, pp. 713–727, 2019.

[18] J. Maslan and L. Cicmanec, "A system for the automatic detection and evaluation of the runway surface cracks obtained by unmanned aerial vehicle imagery using deep convolutional neural networks," *Applied Sciences*, vol. 13, no. 10, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/10/6000

[19] A. Malekloo, X. C. Liu, and D. Sacharny, "Ai-enabled airport runway pavement distress detection using dashcam imagery," *Computer-Aided Civil and Infrastructure Engineering*, vol. n/a, no. n/a. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.13200

[20] J. Jeong, Y. Zou, T. Kim, D. Zhang, A. Ravichandran, and O. Dabeer, "Winclip: Zero-/few-shot anomaly classification and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 606–19 616.

[21] Q. Zhou, G. Pang, Y. Tian, S. He, and J. Chen, "Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection," 10 2023. [Online]. Available: http://arxiv.org/abs/2310.18961

[22] G. Yong, K. Jeon, D. Gil, and G. Lee, "Prompt engineering for zero-shot and few-shot defect detection and classification using a visual-language pretrained model," *Computer-Aided Civil and Infrastructure Engineering*, vol. 38, no. 11, pp. 1536–1554, 2023. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12954

[23] Y. Cao, X. Xu, C. Sun, Y. Cheng, Z. Du, L. Gao, and W. Shen, "Segment any anomaly without training via hybrid prompt regularization," *arXiv preprint arXiv:2305.10724*, 2023.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[25] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[26] 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*, 2018.

[27] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[28] M. Krestenitis and K. Ioannidis, "D3.1 visual analysis for real sensing," Oct. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7220100

[29] R. Chacón, C. Ramonell, and H. Posada, "Deliverable 5.2 " Digital-Twin Enabled multi- physics simulation and model matching"," May 2023. [Online]. Available: https://doi.org/10.5281/zenodo.7928384

[30] I. Stipanovic, S. S. Palic, E. Ganic, and M. Darwish, "D5.4 GIS FOR DIGITALLY TWINNED ASSET MANAGEMENT," 2024. [Online]. Available: https://doi.org/10.5281/zenodo.12772481

[31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[32] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 19–27.