CNN BASED MULTI-LABEL IMAGE CLASSIFICATION FOR PRESENTATION RECOMMENDER SYSTEM

Maria Vlahova–Takova, Milena Lazarova*

Department Computer Systems, Faculty of Computer Systems and Technologies,
Technical University of Sofia, Sofia,
Bulgaria

* Corresponding Author, e-mail: mariavlahova.92@gmail.com

Abstract: Recommender systems are super popular nowadays for giving personalized recommendations to users. The paper focuses on a special type recommender system aimed to advise an improvement of a user's presentation based on slides classification. Classification is a fundamental task in information extraction and retrieval with a goal to assign given resource to a predefined class. In some cases, as in slides' categorization, resources can belong to multiple classes, leading to a multi-labeled classification problem. Two different approaches for solving a multi-labeled image classification problem for the purpose of a presentation recommender system are suggested and evaluated in the paper. They are based on problem transformation and algorithm adaptation strategies and utilize a convolutional neural network for model training.

Key words: multi-labelled classification, image classification, convolutional neural network, recommender system.

1. INTRODUCTION

Classification is a broadly adopted technique in data mining, statistics, machine learning, genetic algorithms, and neural networks. It involves a two-step process: constructing a classification model by analyzing a training dataset characterized by specific attributes, and then, applying this model to categorize new samples whose class labels are unknown.

In machine learning, multi-label classification is gaining significant importance in fields like image recognition, text processing, and biology [1]. Unlike traditional classification, where each item is associated with a single label, multi-label classification assigns multiple labels to a single item, increasing the complexity and necessitating more sophisticated methods. This approach is particularly crucial when labels are interrelated, as in image classification, where a single image may fall into multiple categories simultaneously, or when there's a need to rank these categories. It also applies when no label applies to a given sample. The complexity of multi-label classification grows exponentially as more labels are added, making it increasingly difficult to manage.

Traditional classification methods often struggle in these scenarios, as they may predict labels that are individually plausible but collectively inconsistent or unrealistic. Additionally, applying penalty techniques to enhance model diversity in this context presents further challenges [2, 3].

Recommendation systems help users deal with information overload by predicting which items a user might like and creating a ranked list of recommendations. Many platforms use these systems to offer personalized suggestions that match user needs. Recommendation systems are categorized into collaborative filtering, content-based, utility-based, demographic-based, knowledge-based, and hybrid-based [4]. The most commonly used filtering approaches are content-based and collaborative filtering [4, 5]. In [5] a collaborative friendship recommendation system is proposed based on Neural Collaborative Filtering (NCF). The authors in [6] suggest a performance aware content based image retrieval system that uses grey-level co-occurrence matrix (GLCM). Nowadays recommender systems are aimed at different applications and are widely used in many business cases that utilize user's suggestions of various media types (audio, text, images, videos) [7].

A presentation advisor is a recommender system whose goal is to advise an improvement of a user's presentation. The presentation improvement comprises suggestions and advice relevant to the presentation content. Thus, the presentation recommender systems require a foremost understanding of the presentation design that could further reveal some weaknesses and options for improvement. To accomplish that goal the presentation slides are regarded as images that are categorized into one or several predefined categories.

The goal of the multi-label classification for presentation recommendation system is to classify presentation slides to one or more predefined categories taking into account that categories might also be related. An additional problem in multi-label classification for presentation recommender system is the requirement for relevant dataset to be used for model training [8].

This paper explores two different strategies for solving a multi-label image classification problem. The two approaches are based on problem transformation and algorithm adaptation methods. The solutions is be designed to address the specific difficulties associated with classifying image slides and utilizes convolutional neural networks to train the models effectively. A custom dataset of 912 manually annotated images is used, each representing a presentation slide. The slides are categorized into various types, such as bullet points, slides with interesting content, readable slides, agenda slides, slides with appropriate text size, slides with tables, well-positioned elements, pictures, infographics, consistency in design, and those with graphics [8].

2. RELATED WORK

Classification problems in machine learning are generally regarded as single-label and multi-label classification problems. In addition, single-label classification problems are divided into binary and multi-class types. Binary classification assigns data to one of two classes, while multi-class classification assigns data to one of several possible labels. [9].

Two main approaches are used for multi-class classification: one-versus-one and one-versus-all. A one-versus-one strategy trains as many classifiers as there are pairs of labels. One-versus-all, also referred to as one-vs-rest, is a strategy for multi-class classification aimed to classify instances into one of N classes by creating N binary classifiers, each focused on distinguishing one specific class from all other classes, thus for N classes, this approach generates N binary classifiers [10, 11].

In addition, a modification of the one-versus-all strategy is the error-correcting output code (ECOC) [11] that preprocesses the multi-class target class label into a binary code (an array of 0s and 1s). In this method, each class in a dataset is represented by a unique codeword, which is a row in a coding matrix MMM. The matrix MMM typically consists of binary or ternary values, corresponding to binary or ternary classifiers, respectively. The ECOC process has two main steps: encoding and decoding. During encoding, each class is assigned a codeword, and the classifiers are trained on these encoded labels. In the decoding step, the predicted labels are compared to the codewords, and the class with the codeword closest to the predicted vector, based on Hamming distance, is selected as the output. If the dimension of bit encoding is kept less than the cardinality of the target class label, then it trains comparatively fewer estimators than the one-vs-rest classifier. In theory, log2(n_classes) is sufficient to represent the target class label unambiguously [12].

In multi-label classification, the aim is to predict one or more classes for each input, allowing it to be assigned multiple labels simultaneously. Unlike traditional classification, these labels are not mutually exclusive. This approach is often used in fields like natural language processing, where a single text might address multiple topics, or in image analysis, where an image could contain several distinct objects. To tackle these challenges, three main strategies are commonly used: algorithm adaptation, problem transformation, and ensemble methods [13].

- ***Algorithm adaptation approach***: The algorithm adaptation approach modifies existing algorithms like Boosting, kNN, Decision Trees, Neural Networks, and SVM to handle multi-label classification [9].

- ***Problem transformation approach***: The problem transformation approach converts a multi-label problem into multiple binary or multi-class problems, using single-label classifiers and combining their results for multi-label classification [9].

- ***Ensemble methods***: The ensemble methods use an ensemble of algorithm adaptation and problem transformation methods and combine the results to perform multi-label classification [2].

Imbalanced classification refers to a classification task where the distribution of examples across different classes is uneven in the training data [11]. Several techniques are used to balance the distribution of the original dataset by using different resampling techniques. Multi-Label Random Resampling uses undersampling (randomly deleting examples in the majority class) and oversampling (randomly duplicating examples in the minority class) [11, 14]. Multi-Label Heuristic Resampling selects instances to delete, or clone based on specific rules rather than randomly thus aiming to minimize information loss during undersampling and overfitting during oversampling [11, 3].

A thorough review and experimental evaluation of the multi-label classification methods is presented in [2]. The authors discuss different approaches for multi-label classification and give an empirical evaluation of 26 predictive models for 42 datasets in various domains, including text (news, reports), medicine, multimedia (images and audio), bioinformatics, biology and chemistry concluding that random forest and neural network based approach give best performance among the studied algorithm adaptation methods.

The comprehensive study of multi-label classification approaches in [15] covers extreme multi-label classification, multi-label learning with limited supervision, deep learning for multi-label learning, online multi-label learning, and statistical multi-label learning. The discussion of the utilization of deep learning for multi-label image classification focuses on a review of using Recurrent Neural Networks (RNN), Convolutional Neural Networks, (CNN), Long-Short Term Memory (LSTM), Graph Convolutional Network (GCN). Transformers are also utilized to solve multi-label image classification. The authors in [16] present a Multi-label Transformer architecture (MlTr) that adopts a transformer module in CNN-based architecture. A novel framework Classification Transformer (C-Tran) is suggested in [4] to better solve multi-label image classification by using a label mask during the training.

3. METHODOLOGY FOR MILTI-LABEL IMAGE CLASSIFICATION IN PRESENTATION RECOMMENDER SYSTEM

3.1. Problem Transformation Approach for Milti-Label Image Classification in Presentation Recommender System

For designing multi-label image classification features within a presentation recommendation platform, a strategy based on the problem transformation approach is utilized by training several binary classificators to predict each label. In the proposed approach, multiple binary classifiers are trained independently, each focusing on a single label in a multi-label image dataset. Each classifier determines whether a particular label is present in a given instance. The results from all binary classifiers are then combined to determine the final multi-label outcome for each instance. The processing steps of the suggested problem transformation approach for multi-label image classification are given in Algorithm 1 (Fig. 1).

Initially, the dataset is loaded, formatted correctly and the images undergo preprocessing which includes resizing and color adjustment to RGB, followed by feature extraction using a ResNet50 model to generate embeddings. These embeddings are analyzed for visual uniqueness using cosine similarity to avoid duplicates and ensure diversity. An original undersampling strategy, developed specifically for this study, is applied to balance the dataset by reducing the frequency of over-represented classes and minimizing model bias. Finally, the dataset is equally divided across all categories and split into training, validation, and test sets, ready for model training and evaluation. As the dataset is not large enough, cross-validation cannot be employed effectively and is not included in the processing steps. This process ensures a balanced, diverse dataset that enhances model accuracy and generalization.

Algorithm 1: Multi-label image dataset processing

```
# Step 1: Load Dataset
LOAD dataset containing image file paths and multi-label annotations
REMOVE any duplicate images based on file paths and labels
# Step 2: Data Preprocessing
FOR each image in dataset:
    LOAD image from file path
    RESIZE image to the required dimensions
    CONVERT image to RGB format
# Step 3: Data undersampling
EXTRACT class labels for each image
GROUP images by class label to identify majority classes
FOR each class in the majority classes:
    EXTRACT embeddings for all images in the class using ResNet50 model
    COMPUTE pairwise cosine similarities between embeddings

    CALCULATE average cosine similarity score for each image
    SORT images based on their average similarity score
    SELECT visually distinct images with low cosine similarity scores
# Step 4: Remove Duplicate or Near-Duplicate Images
FOR each image in dataset:
    IF image has high cosine similarity with others (near-duplicate):
      REMOVE image from dataset
# Step 5: Dataset Balancing
FOR each class in dataset:
    CALCULATE number of samples needed to balance all categories
    RANDOMLY SELECT visually distinct images for each class
# Step 6: Dataset Splitting
    SPLIT balanced dataset into training, validation set and test set
# Step 7: Train CNN Model
DEFINE CNN model with the proposed architecture
COMPILE model with:
     Loss Function = 'binary_crossentropy',
     Optimizer = 'adam',
     Metrics = ['accuracy']
# Step 8: Train and Evaluate the Model
TRAIN the model using the training set
VALIDATE the model using the validation set
EVALUATE the model on the test set
# Final Output: Trained CNN model ready for predictions
RETURN trained CNN model
```

Figure 1. Processing steps of the problem transformation approach for multi-label classification

At the model training step, various machine learning algorithms are tested to identify the most effective one for the task of multi-label image classification for the presentation recommender system. The models evaluated include Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbours, Naive Bayes, Back Propagation Neural Network (BPNN), and Convolutional Neural Network (CNN). Among these, the CNN demonstrated the best performance and thus it is selected for model training for the multi-class presentation recommender system.

The suggested CNN architecture utilized to train a model for the described presentation recommender system is presented in Fig. 2. The architecture consists of four

convolutional blocks, two fully connected layers, and a final output dense layer. Each convolutional block includes a convolutional layer, which is used to detect features such as edges, textures, and more complex patterns, followed by batch normalization to stabilize and accelerate the training process.ReLU activation is applied to add non-linearity, enabling the model to capture more complex patterns, and max pooling is used to reduce spatial dimensions while preserving essential features. The fully connected layers contain dense layers, followed by batch normalization, ReLU activation, and dropout to enhance model generalization and prevent overfitting. The final layer is a single neuron because it is designed for binary classification.

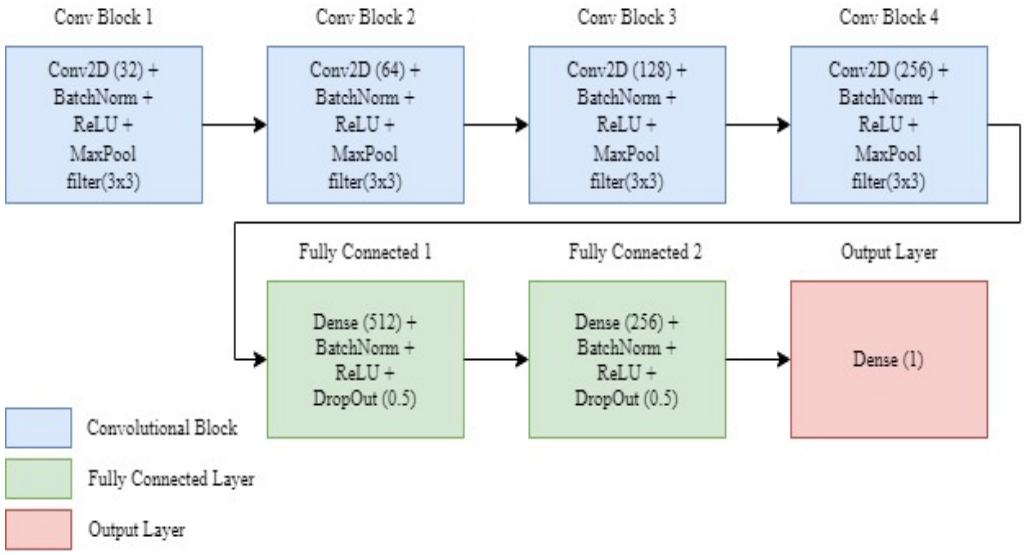

Figure 2. CNN architecture used for the multi-label image classification for presentation recommender system

3.2. Algorithm Adaptation Approach for Multi-Label Image Classification in Presentation Recommender System

To adopt the algorithm adaptation approach for multi-label image classification in the presentation recommender system, the strategy involves utilizing a multi-labelled image dataset to develop a single model that can accommodate all labels simultaneously. Although this approach is susceptible to overfitting, it could effectively consider the relationships and dependencies among labels, offering a comprehensive model that captures the inherent complexities of multi-labelled data. To address the imbalance in the dataset, suitable data preprocessing methods are used. The data preprocessing for the algorithm adaptation approach follows the following steps 1÷6 as presented in Algorithm 1. At the model training (step 7) a CNN is used with the architecture as given in Fig. 2, only the final layer is designed to output probabilities for each label, indicating the likelihood of each label being present.

4. RESULTS AND DISCUSSION

4.1. Experimental setup

• ***Dataset*:** The dataset used for the experimental evaluation of the proposed approaches for multi-label image classification in the presentation recommender system consists of 912 manually annotated images that represent presentation slides. The experimental approach involves modelling, where we build and train machine learning models to classify images based on their features. The split ratios used for the model training and evaluation are 70% for training, 10% for validation, and 20% for testing. This ensures enough data to train the model, validate its performance, and evaluate its generalization ability on unseen data. Unfortunately, the dataset is insufficient for cross-validation to be included.

The images are categorized into 10 classes that represent certain features in each image slide according to its usage for the presentation recommender system: slides with bullets, interesting slides, readable slides, agenda slides, slides with the proper size of the text, slides with tables, slides with proper positioning, slides with pictures, infographics, and graphics. Each label is a binary indicator '0' or '1', indicating whether an image contains or lacks a particular feature. The distribution of the labels in the image datasets is given in Table 1.

The correlation matrix (Table 2) provides insights into the relationships between the classes in the multi-label image dataset. Each entry in the matrix is the value of the Pearson correlation coefficient between two labels, ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation). As can be seen, high positive correlations exist between the labels "readable" and "size_of_text", "readable" and "positioning", "interesting" and "infographics", "interesting" and "graphics" and negative is the correlation between "pictures" and "infographics", "bullets" and "size_of_text".

• ***Parameter settings*:** In order to evaluate the suggested approaches, the presented data processing methodology and model training are implemented using Python and Tensorflow. The data processing and models' training is done on using the following experimental platform: Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz, 64 GB RAM. GridSearchCV is utilized to optimize the CNN and perform hyperparameter tuning. The best parameters identified are given in Table 3.

• ***Evaluation metrics*:** Classification evaluation is based on the widely used evaluation metrics accuracy, precision, recall, F1 score. In addition Hamming Loss, Jaccard similarity, Kappa coefficient, ROC AUC and training time are also measured and compared [17].

Table 1. Label distribution of the experimental multi-label image dataset

Label	*Number of images (%)*	*Label*	*Number of images (%)*
L1: bullets	*15.8*	*L6: tables*	*4.9*
L2: interesting	*39.7*	*L7: positioning*	*79.8*
L3: readable	*62.2*	*L8: pictures*	*67.5*
L4: agenda	*0.7*	*L9: infographics*	*22.0*
L5: size of text	*58.7*	*L10: graphics*	*37.5*

Table 2. Correlation matrix for the experimental multi-label image dataset

Label	*L1*	*L2*	*L3*	*L4*	*L5*	*L6*	*L7*	*L8*	*L9*	*L10*
L1	*1.0000*	*0.0222*	*-0.0291*	*0.0027*	*-0.1272*	*-0.0305*	*-0.1237*	*-0.0016*	*-0.1787*	*0.0742*
L2	*0.0222*	*1.0000*	*0.0403*	*-0.0372*	*0.1713*	*-0.0536*	*0.1865*	*0.0575*	*0.1983*	*0.1493*
L3	*-0.0291*	*0.0403*	*1.0000*	*0.0664*	*0.3946*	*-0.0097*	*0.2920*	*-0.1582*	*0.1271*	*-0.0252*
L4	*0.0027*	*-0.0372*	*0.0664*	*1.0000*	*-0.0630*	*-0.0192*	*0.0422*	*0.0262*	*0.0417*	*-0.0335*
L5	*-0.1272*	*0.1713*	*0.3948*	*-0.0631*	*1.0000*	*-0.0274*	*0.0925*	*0.0330*	*0.0227*	*-0.1115*
L6	*-0.0305*	*-0.0535*	*-0.0097*	*-0.0192*	*-0.0274*	*1.0000*	*-0.0181*	*-0.1162*	*-0.0330*	*-0.1055*
L7	*-0.1237*	*0.1865*	*0.2920*	*0.0422*	*0.0925*	*-0.0181*	*1.0000*	*0.0069*	*0.1604*	*0.1351*
L8	*-0.0016*	*0.0575*	*-0.1582*	*0.0262*	*0.0330*	*-0.1162*	*0.0069*	*1.0000*	*0.0232*	*-0.1861*
L9	*-0.1787*	*0.1983*	*-0.1271*	*-0.0417*	*0.0227*	*-0.0330*	*0.1604*	*0.0232*	*1.0000*	*0.1931*
L10	*0.0743*	*0.1493*	*-0.0252*	*-0.0335*	*-0.1115*	*-0.1055*	*0.1351*	*-0.1861*	*0.1931*	*1.0000*

Table 3. CNN hyper parameters

Parameter	***Value***		***Parameter***	***Value***
Batch size	*16*		Convolution layers	2
Dropout rate	0.0		Dense units	128
Training epochs	200		Filters	64
Learning rate	0.001			

4.2. Model evaluation for problem transformation approach

The evaluation results based on the selected metrics of the suggested approach for problem transformation in multi-label image classification in the presentation recommender system are given in Table 4. The results for the accuracy, precision, recall, and F1-score metrics point out the usability of the suggested approach. The Kappa coefficient results of 0.5253 provide an evaluation of the classification as "moderate agreement" according to [15]. The obtained accuracy for each class is given in Table 5. As can be seen, the accuracy is over 0.7 for each class with the highest values obtained for "positioning" and "graphics" labels (0.8556 and 0.8493 respectively).

Table 4. Evaluation metrics for the problem transformation method

Metric (average)	***Value***	***Metric (average)***	***Value***
Hamming Loss	*0.2373*	*Jaccard similarity*	*0.5790*
Accuracy	*0.7627*	*Kappa coefficient*	*0.5253*
Precision	*0.8137*	*ROC AUC*	*0.7622*
Recall	*0.7164*	*Training time*	*37 min*
F1-Score	*0.7889*		

Table 5. Accuracy per class for the problem transformation method

Label	***Accuracy***	***Label***	***Accuracy***
L1: bullets	*0.7213*	*L6: tables*	*0.7557*
L2: interesting	*0.8379*	*L7: positioning*	*0.8556*
L3: readable	*0.7378*	*L8: pictures*	*0.7874*
L4: agenda	*0.8261*	*L9: infographics*	*0.7007*
L5: size_of_text	*0.7935*	*L10: graphics*	*0.8493*

4.3. Model evaluation for algorithm adaptation approach

The evaluation results based on selected metrics of the suggested approach for algorithm adaptation for multi-label image classification in the presentation recommender system are given in Table 6 and the obtained accuracy for each class is given in Table 7. This approach shows moderate performance, with an accuracy of 0.7306 and a Jaccard similarity of 0.5511. There is a noticeable difference between Precision and Recall, leading to lower F1 scores, especially in the macro-averaged metrics. The algorithm adaptation approach reveals significant variability in class-specific accuracy, with some classes performing much better than others. Overall, the model shows moderate reliability with a Kappa coefficient of 0.4459 and an ROC AUC of 0.6118, indicating room for improvement. The results for the accuracy, precision, recall, and F1-score metrics point out the usability of the suggested approach even if the results are worse than in the previously reported model.

Table 6. Evaluation metrics for the problem transformation approach

Metric (average)	***Value***	***Metric (average)***	***Value***
Hamming Loss	*0.2918*	*Jaccard similarity*	*0.5511*
Accuracy	*0.7306*	*Kappa coefficient*	*0.4459*
Precision (Macro)	*0.6013*	*Precision (Micro)*	*0.6404*
Recall (Macro)	*0.5297*	*Recall (Micro)*	*0.7146*
F1-Score (Macro)	*0.4914*	*F1-Score (Micro)*	*0.6755*
Training time	*21.87 seconds*	*ROC AUC*	*0.6118*

Table 7. Accuracy per class for the algorithm adaption approach

Label	***Accuracy***	***Label***	***Accuracy***
L1: bullets	*0.8092*	*L6: tables*	*0.7557*
L2: interesting	*0.4656*	*L7: positioning*	*0.6718*
L3: readable	*0.5573*	*L8: pictures*	*0.8092*
L4: agenda	*0.6318*	*L9: infographics*	*0.7328*
L5: size_of_text	*0.9542*	*L10: graphics*	*0.6183*

4.4. Discussion

While several studies have addressed classification and multi-label classification problems, a direct comparison of the results with the work presented in the paper is not feasible due to the use of a custom dataset.

The dataset used for the evaluation shows low correlation between classes, and as expected, the problem transformation approach outperforms the algorithm adaptation approach in most evaluation metrics, such as Hamming loss, accuracy, precision, recall, F1-score, and ROC AUC, indicating better overall classification performance. However, this approach comes with a significant drawback: it is computationally expensive, requiring 37 minutes for model training compared to just 22 seconds for the algorithm adaptation approach.

The problem transformation approach has several key advantages. It is simple, as each binary classifier is straightforward to implement and train. It offers flexibility,

allowing different algorithms to be applied to individual labels based on their unique traits. This approach is also scalable, since adding a new label only requires training a new classifier without affecting existing ones. Furthermore, it is easier to interpret, as each classifier is responsible for a single label, making it easier to debug. Balancing the dataset for each label separately is also more manageable, allowing for more effective handling of class imbalance. However, it also has some drawbacks. It is computationally costly, as managing a large number of classifiers requires substantial resources. Additionally, it assumes that the labels are independent, which may not hold true in practice, potentially missing important interdependencies between labels. Moreover, since classifiers are trained independently, there may be inconsistent predictions across labels, such as predicting mutually exclusive labels together.

On the other hand, the algorithm adaptation approach has its own advantages. It directly addresses the multi-label nature of the problem, preserving relationships and dependencies between labels. This approach provides a unified framework for handling multiple labels, simplifying both training and inference. It comes with the cost of scalability issues, risk of overfitting, increased complexity and problems with interpretability.

In summary, while the problem transformation approach excels in performance, it comes at a higher computational cost. The algorithm adaptation approach is more efficient but struggles with scalability and interpretability, making it less suitable for applications with a large number of labels or complex datasets.

5. CONCLUSION

The suggested approaches for CNN-based multi-label image classification for presentation recommender system are based on the two widely used strategies for problem transformation and algorithm adaptation, for addressing multi-label classification challenges. While inspired by established theoretical frameworks, the algorithm developed for this study is custom-built, specifically tailored to handle the unique aspects of presentation slides as a dataset. In both approaches the utilized data processing pipeline involves dataset preprocessing for feature extraction and dataset balancing by undersampling to prevent model overfitting. The experimental evaluation is based on a custom image dataset with presentation slides annotated with ten labels that exhibit a high correlation between several of the categories. The performance analyses of the two suggested strategies reveal that the problem transformation approach excels in most performance metrics but this improved performance comes at a significant computational cost with training times much longer compared to the algorithm adaptation strategy. On the other hand, the algorithm adaptation approach, while slightly lagging in some performance metrics, demonstrates remarkable efficiency, requiring significantly less computational time and resources. In both experimental setups, the image dataset used for model training and evaluation is relatively small due to the lack of large public image datasets that can be used for presentation recommender systems, Furthermore, the exploratory data analysis of the dataset revealed minimal correlations among labels, indicating that a problem transformation approach which treats each label separately may be more effective for our case than using an algorithm adaptation

approach. For applications demanding high accuracy and where computational resources are abundant. Future studies on multi-label image classification in the context of presentation recommender systems will focus on developing a hybrid approach that combines the strengths of both strategies in an attempt to achieve a balance between performance and efficiency.

ACKNOWLEDGEMENT

The research work presented in the paper is funded by European Union-NextGenerationEU via the National Recovery and Resilience Plan of the Republic of Bulgaria under project BG-RRP 2.004-0005 "Improving the research capacity and quality to achieve international recognition and reSilience of TU-Sofia (IDEAS)".

REFERENCES

[1] Dhatri, G., Rachana, B. A survey on multi-label classification. *Recent Trends in Programming Languages*, vol. 5, no. 1, 2018, pp. 19-23.

[2] Bogatinovski, J., Todorovski, L., Džeroski, S., Kocev, D., Comprehensive comparative study of multi-label classification methods. *Expert Systems with Applications,* Volume 203, 2022, DOI: https://doi.org/10.1016/j.eswa.2022.117215.

[3] Katona, T., Tóth, G., Petró, M., Harangi, B. Advanced Multi-Label Image Classification Techniques Using Ensemble Methods. *Machine Learning and Knowledge Extraction*, 2024, pp. 1281–1297. DOI: https://doi.org/10.3390/make6020060.

[4] Ko, H., Lee, S., Park, Y., Choi, A. A survey of recommendation systems: recommendation models, techniques, and application fields. *Electronics*, vol. 11, no. 1, 2022, DOI: https://doi.org/10.3390/electronics11010141.

[5] Alshammari, A., Alanazi, R., A neural network-based collaborative filtering model for social recommendation systems. *International Journal on Information Technologies and Security,* vol.16, no.3, 2024, pp. 27-36, DOI: https://doi.org/10.59035/ILMO8300.

[6] Battur, R., Jagadisha N. A performance aware contents-based image retrieval (CBIR) technique. *International Journal on Information Technologies and Security*, vol.14, no.2, 2022, pp. 87-98.

[7] Fayyaz, Z., Ebrahimian, M., Nawara, D., Ibrahim, A., Kashef, R. Recommendation systems: algorithms, challenges, metrics, and business opportunities. *Applied Sciences*, vol. 10, 2020, DOI: https://doi.org/10.3390/app10217748.

[8] Vlahova, M., Lazarova, M. Collecting a custom database for image classification in recommender systems. *2022 IEEE 10th International Scientific Conference on Computer Science (COMSCI)*, Sofia, Bulgaria, 2022, October 2022, DOI: 10.1109/COMSCI55378.2022.9912591.

[9] Er, M., Venkatesan, R., Wang, N. An online universal classifier for binary, multi-class and multi-label classification. *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, Hungary, October 2016, pp. 3701-3706, DOI: 10.1109/SMC.2016.7844809.

[10] Pawara, P., Okafor, E., Groefsema, M., Sheng, H., Schomaker, L., Wiering M., One-vs-One classification for deep neural networks. *Pattern Recognition*, vol. 109, 2020, ISSN 0031-3203, DOI: https://doi.org/10.1016/j.patcog.2020.107528.

[11] Gao, X., He, Y., Zhang, M., Diao, X., Jing, X., Ren, B., Ji, W. A multiclass classification using one-versus-all approach with the differential partition sampling ensemble. *Engineering Applications of Artificial Intelligence,* vol. 97, 2021, DOI: https://doi.org/10.1016/j.engappai.2020.104034.

[12] Kim, J., Kim, M., Error-correcting output codes for multi-class classification based on Hadamard matrices and a CNN model. *Procedia Computer Science*, vol. 222, 2023, pp. 252-261, DOI: https://doi.org/10.1016/j.procs.2023.08.163.

[13] Tarekegn, A., Giacobini, M., Michalak, K. A review of methods for imbalanced multi-label classification. *Pattern Recognition*, vol. 118, 2021, DOI: https://doi.org/10.1016/j.patcog.2021.107965.

[14] Taha, A. Y., Tiun, S., Rahman, A. H. A., Sabah, A., Mulilabel over-sampling and under-sampling with class alignment for imbalance multilabel text classification. *Journal of Information and Communication Technology*, vol. 20, no. 3, 2021, DOI: https://doi.org/10.32890/jict2021.20.3.6

[15] Liu, W., Wang, H., Shen, X., Tsang, I. The emerging trends of multi-label learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, 2022, pp. 7955-7974, DOI: 10.1109/TPAMI.2021.3119334.

[16] Lanchantin, J., Wang, T., Ordonez, V., Qi, Y. General multi-label image classification with transformers. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nashville, TN, USA, 2021, pp. 16473-16483, DOI: 10.1109/CVPR46437.2021.01621.

[17] Vujovic, E., Classification model evaluation metrics, *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, 2021, DOI: https://doi.org/10.14569/ijacsa.2021.0120670.

Information about the authors:

Maria Vlahova–Takova – PhD student, MSc., Eng., Technical University of Sofia, Dept. Computer Systems, Faculty of Computer Systems and Technologies. Research interests: artificial intelligence, machine learning, deep learning, image analyses, recommender systems.

Milena Lazarova – Professor, PhD, MSc. Eng., Technical University of Sofia, Dept. Computer Systems, Faculty of Computer Systems and Technologies. Research interests: artificial intelligence, machine learning, deep learning, computer vision, metaheuristics.

Manuscript received on 18 August 2024