Leveraging Diverse CNN Architectures for Medical Image Captioning: DenseNet-121, MobileNetV2, and ResNet-50 in ImageCLEF 2024

Notebook for the VIT_Conceptz Lab at CLEF 2024

Sriram Ram[1,*], Shashaank Vinoth[1,*], Rahul Natesh Gopalakrishnan[1,*], Aastick Amirteswar Balakumar[1,*], Lekshmi Kalinathan[1,*] and Thomas Abraham Joseph Velankanni[1,*]

[1]*Vellore Institute of Technology, Chennai Campus, Vandalur-Kelambakkam Road, Chennai, Tamil Nadu 600127, India*

Abstract

In this study, we employed three deep learning models ResNet50, MobileNetV2, and DenseNet-121 to perform concept detection, which involves identifying and locating relevant concepts in medical images. This provides the foundation for generating coherent captions in the subsequent caption prediction. In medical imaging, concept detection plays a pivotal role. It enables accurate disease diagnosis and monitoring by identifying specific features, such as tumors, fractures, or anomalies. These concepts guide treatment planning, ensuring timely interventions. Among the models, ResNet50 achieved the highest performance, followed by MobileNetV2 and DenseNet-121. These results indicate that ResNet50 is the most effective model for identifying relevant concepts within medical images. This study provides insights into the applicability of different convolutional neural networks for medical image analysis, contributing to advancements in automated medical image captioning. Our team secured 8th place on the overall challenge leaderboard in the Concept Detection Task of the 8th edition of Caption Challenge in ImageCLEFmedical 2024.

Keywords

ImageCLEF 2024, Image Captioning, DenseNet121, Resnet50, MobileNetV2,

1. Introduction

The 8th edition of the Caption Challenge in the ImageCLEFmedical 2024 [1] focuses on two tasks: Concept Detection and Caption Prediction [2]. This study examines the results derived from pre-trained convolutional models—ResNet50, MobileNetV2 and DenseNet-121—utilized specifically for the Concept Detection task. Concept Detection involves multilabel classification, where each radiology image may contain one or more labels. These labels, represented as CUIs (Controlled User Information), are mapped to specific concepts. Identifying these concepts helps in isolating individual components within the image and can be further applied to information retrieval. The motivation for this task stems from the growing availability of images without accompanying metadata. Acquiring metadata is crucial for making the content usable and accessible for further analysis and application [3].

Previous studies have underscored the challenges and potential of various approaches in medical image analysis. Rahman [4] demonstrated commendable precision in lesion detection using a bespoke CNN architecture, highlighting the effectiveness of tailored models in specific tasks. Dimitris and Ergina [5] explored the efficacy of transfer learning in medical imaging, showing that pretrained models can be effectively adapted for diverse imaging modalities. Rossetto et al. [6] extended visual concept detection to video retrieval, showcasing the versatility of these techniques across different formats. Ohri and Kumar [7] presented a comprehensive framework for medical image classification,

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

*Corresponding author.

sriram.r1003@gmail.com (S. Ram); shashaankvinoth@gmail.com (S. Vinoth); rahulnateshgr8@gmail.com (R. N. Gopalakrishnan); aastick.amirteswar.b@gmail.com (A. A. Balakumar); lekshmi.k@vit.ac.in (L. Kalinathan); thomasabraham.jv@vit.ac.in (T. A. J. Velankanni)

suggesting that integrating diverse methodologies can enhance the accuracy and applicability of concept detection.

2. Methodology

The dataset for the concept detection task is derived from the Radiology Objects in COntext Version 2 (ROCOv2) dataset [8], an enhanced version of the original Radiology Objects in COntext (ROCO) dataset [9]. This dataset is specifically curated for radiology images and is sourced from biomedical articles in the PMC OpenAccess subset. The training set comprises 70,108 radiology images, the validation set includes 9,972 images, and the test set contains 17,237 images.

2.1. Dataset Processing and Environmental Setup

We began by loading the train_concepts.csv file to extract UMLS (Unified Medical Language System) [10] concept IDs from the CUIs column. Images were resized to 224x224 pixels, normalized, and converted to the appropriate format. Generators were created for the training and validation datasets to handle multilabel classification, yielding batches of images and corresponding CUIs.

The implementation was configured with Python 3.6 or higher, and the necessary libraries, including TensorFlow(v2.15.0), Keras(v2.15.0), and NumPy(v1.24.3), were installed as shown in the listing below. We also ensured compatibility with CUDA and cuDNN to leverage GPU acceleration for model training, significantly improving computational efficiency. In addition, the default learning rate scheduler used in our experiments is the ReduceLROnPlateau method from TensorFlow/Keras, which reduces the learning rate when a monitored metric, such as validation loss, has stopped improving. We also implemented the EarlyStopping callback to stop training when the validation loss metric shows no improvement for a specified number of epochs. These methods help optimize the training process and prevent overfitting.

```
# Python version
Python 3.6 or higher

# Required libraries and versions
TensorFlow v2.15.0
Keras v2.15.0
NumPy v1.24.3
```

Listing 1: Environment Setup

2.2. DenseNet-121

In this study, we employ the DenseNet-121 architecture for image classification tasks. DenseNet-121 is a densely connected convolutional network that enhances information flow between layers by connecting each layer to every other layer in a feed-forward fashion shown in Fig. 1 [11]. The model begins with an input layer for images of size $224 \times 224 \times 3$, followed by an initial convolutional layer with 64 filters of size 7×7 and a stride of 2. This is followed by a batch normalization layer, a ReLU activation function, and a max pooling layer with a filter size of 3×3 and a stride of 2. The network consists of four dense blocks with increasing complexity. The first dense block contains 6 layers, each comprising two convolutions repeated 6 times. This is followed by a transition layer, which includes a 1×1 convolution and a 2×2 average pooling layer. The second dense block contains 12 layers with two convolutions repeated 12 times, followed by another transition layer. The third dense block consists of 24 layers with two convolutions repeated 24 times, followed by a transition layer. The final dense block has 16 layers with two convolutions repeated 16 times. After dense blocks, the model includes a final batch

normalization layer, a ReLU activation function, an average pooling layer of size 7×7, and a fully connected layer with 1945 output units. This comprehensive design facilitates efficient feature reuse, leading to improved performance and reduced parameter count compared to traditional convolutional networks.

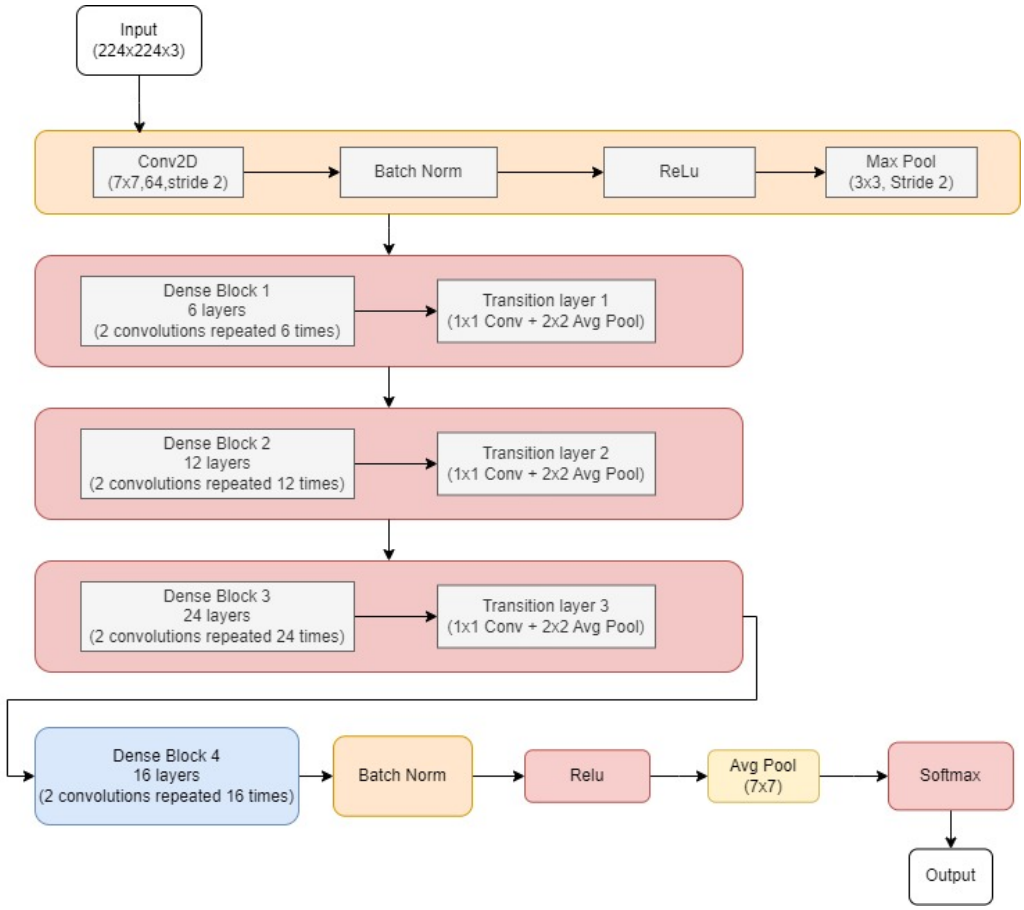

Figure 1: DenseNet-121 Architecture Diagram [12].

2.3. MobileNetV2

In this study, we employ the MobileNetV2 architecture for image classification tasks. MobileNetV2 is a lightweight and efficient convolutional neural network designed for mobile and embedded vision applications shown in Fig. 2. The model begins with an input layer designed for images sized at $224 \times 224 \times 3$ pixels. This is followed by a 3×3 convolutional layer employing 32 filters and a stride of 2, complemented by batch normalization and ReLU activation.
The core of MobileNetV2 consists of inverted residual blocks with linear bottlenecks. Each block includes depthwise convolutions, expansion layers with 1×1 convolutions, and pointwise convolutions to adjust channel dimensions, accompanied by batch normalization and ReLU activation. These blocks are repeated to enhance feature extraction efficiently [13].

After multiple inverted residual blocks, the network uses a global average pooling layer to consolidate spatial information, followed by a flattening layer. The final layers are a fully connected layer and a softmax activation function, which output class probabilities. MobileNetV2's design ensures effective feature extraction with fewer parameters, making it suitable for resource-limited environments.

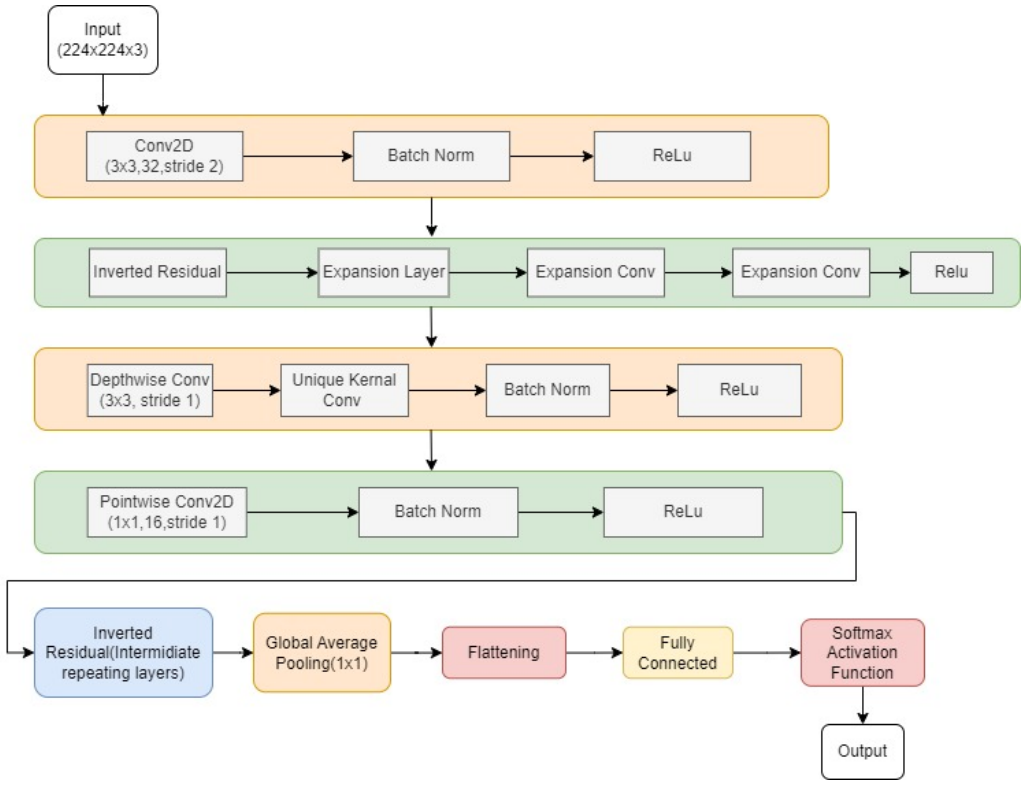

Figure 2: MobileNetV2 Architecture Diagram [13].

2.4. ResNet50

ResNet50 is a deep convolutional neural network architecture consisting of 50 layers, designed to address the vanishing gradient problem through the use of residual blocks shown in Fig. 3. Each block includes skip connections that allow the network to effectively propagate gradients, enabling the training of very deep networks [14]. ResNet50 features a bottleneck design with layers organized into five stages, combining convolutional operations and identity mappings, followed by a global average pooling and a fully connected layer for classification. This architecture achieves high performance on image recognition tasks, making it a cornerstone in modern deep learning.

2.5. Training and Evaluation

In this study, we explore three distinct convolutional neural network architectures—DenseNet-121, MobileNetV2, and ResNet50—for image classification tasks (**Figures 1, 2, and 3**). For **DenseNet-121**, initialized with pre-trained weights from ImageNet and customized with additional dense and classification layers, we conducted initial training with base layers frozen over 20 epochs, using a batch size of 32 and employing data augmentation techniques such as rescaling, horizontal flipping, and zooming. Subsequently, we unfroze the base layers for fine-tuning to adapt the model specifically to our dataset. The model was evaluated on a separate test dataset to assess performance metrics including loss and accuracy, ensuring robustness and reproducibility of results.

MobileNetV2, optimized for efficiency in mobile and embedded applications, was initialized with pre-trained weights and customized with additional dense layers for multi-label classification, aligning with the number of UMLS concepts. The model was compiled using binary cross-entropy loss and the Adam optimizer, with accuracy as the primary evaluation metric. Training commenced with the dataset split into training and validation sets. Over 20 epochs and a batch size of 32, the model underwent initial training with frozen layers, accompanied by rescaling, horizontal flipping, and zooming data augmentation techniques applied solely to the training data for improved generalization. Subsequent

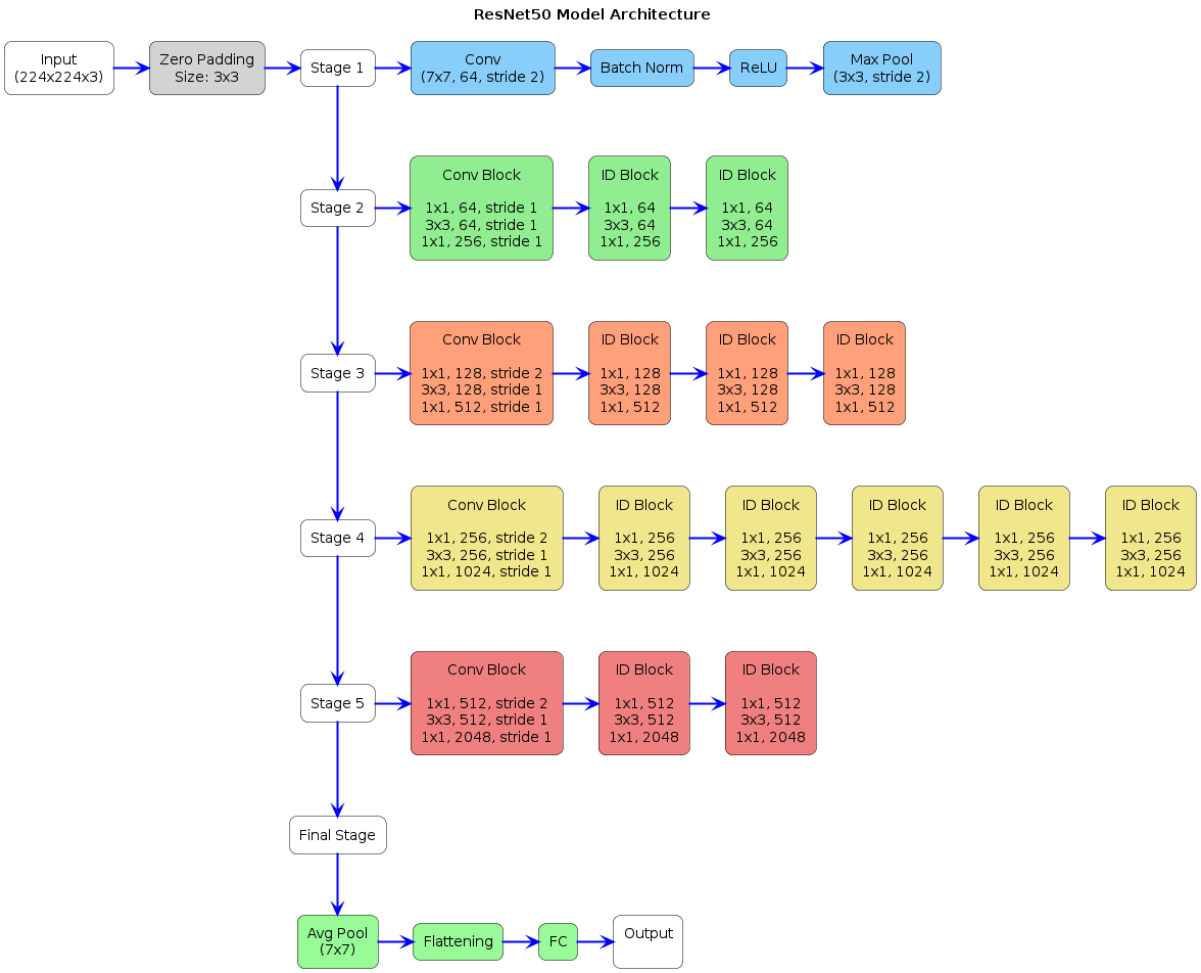

Figure 3: ResNet50 Architecture Diagram [14].

fine-tuning unfroze layers to adapt the model specifically to the dataset. Evaluation on independent validation and test sets verified MobileNetV2's robust performance in classifying images, affirming its suitability for resource-efficient environments and underscoring its effectiveness in medical image analysis tasks.

ResNet50, leveraging pre-trained weights from ImageNet without the top classification layer, was initialized to maintain the integrity of learned features. The base model's layers were frozen to preserve these weights during training. For model compilation, binary cross-entropy served as the loss function for its effectiveness in multilabel classification tasks. The Adam optimizer was chosen to efficiently navigate the model's training process. Accuracy, a critical metric, was employed to evaluate model performance. Training spanned 15 epochs using a batch size of 32, with data augmentation techniques—including rescaling, horizontal flipping, and zooming—applied to enhance generalization capabilities. The model demonstrated robustness and reliability when evaluated on separate validation and test datasets, underscoring its efficacy in complex image recognition scenarios.

3. Experimental Results and Analysis

The results of our analysis are as follows:

Table 1
Comparison of F1 Scores and Secondary F1 Scores for Deep Learning Models.

Model	F1 Score	Secondary F1 Score
ResNet50	0.181	0.264
MobileNetV2	0.178	0.253
DenseNet-121	0.114	0.23

Table 2
Performance Metrics of Different Deep Learning Models

Model	Precision	Recall	F1 Score
ResNet50	0.41	0.36	0.38
MobileNetV2	0.61	0.37	0.46
DenseNet-121	0.35	0.42	0.38

The better F1 score of ResNet50 compared to MobileNetV2 and DenseNet-121 shown in Table (1) can be attributed to its superior performance in reducing loss through the training epochs as observed. The graphs provided display the training and validation accuracy and loss over several epochs for three different models: DenseNet-121 (Fig. 6), MobileNetV2 (Fig. 5), and ResNet-50 (Fig. 4). These visualizations highlight distinct performance characteristics and potential issues like overfitting.

DenseNet-121 (Fig. 6) shows a steady increase in training accuracy, but its validation accuracy fluctuates significantly, indicating instability. The training loss decreases consistently, reflecting effective learning on the training data. However, the validation loss initially decreases and then starts to increase, a clear sign of overfitting. This suggests the model is too complex, capturing noise and nuances that don't generalize well.

In contrast, MobileNetV2 (Fig. 5) demonstrates both training and validation accuracies that increase and converge closely, indicating consistent improvement without significant divergence. The training and validation loss curves also decrease and stabilize, suggesting that the model generalizes well to the validation data. This implies that MobileNetV2, designed for efficiency, maintains a good balance between complexity and generalization, effectively preventing overfitting.

ResNet-50 (Fig. 4) exhibits mild overfitting, with small gaps between training and validation accuracy and slight fluctuations in validation loss, indicating it captures some noise but still generalizes relatively well. With some hyperparameter tuning and regularization, its performance could improve.

In addition, Table 2 presents the precision, recall, and F1 scores for three models: ResNet50, MobileNetV2, and DenseNet-121. ResNet50 has a precision of 0.41, recall of 0.36, and an F1 score of 0.38, indicating moderate performance with a tendency to miss actual positives. MobileNetV2 shows the highest precision at 0.61 and an F1 score of 0.46, suggesting it balances precision and recall better than the other models, despite a recall of 0.37. DenseNet-121 has the highest recall at 0.42 but the lowest precision at 0.35, resulting in an F1 score of 0.38, similar to ResNet50. This indicates that DenseNet-121 identifies more actual positives but also has a higher rate of false positives. Consequently, MobileNetV2 demonstrates the most balanced performance, making it potentially the most reliable model when

considering both precision and recall.

The results achieved by our models, compared to EfficientNet-B0, EfficientNet-v2-s models, and other challenge participants, are notably inferior due to several factors. DenseNet-121 exhibits unstable validation accuracy and increasing validation loss, indicating overfitting and inability to generalize effectively. MobileNetV2, in contrast, demonstrates consistent improvement in both training and validation metrics, indicating better generalization capabilities. ResNet-50 shows mild overfitting with small accuracy gaps and fluctuating validation loss. Additionally, while MobileNetV2 achieves the highest precision and balanced F1 score (0.46), DenseNet-121's high recall comes at the cost of lower precision and similar overall F1 score (0.38) to ResNet-50, indicating challenges in correctly identifying positives and minimizing false positives(Table 2). These performance differences might also stem from the models' regularization techniques, the quality and quantity of the training data, and the tuning of hyperparameters such as learning rates and batch sizes. Enhancing regularization, tuning hyperparameters, or augmenting the dataset could help improve DenseNet-121 and ResNet-50's performance to reduce overfitting and enhance generalization. To ensure reproducibility, the code and model weights for our experiments are accessible on GitHub at **https://github.com/Sriram0703/ImageCLEFmedical-2024-Concept-Detection**.

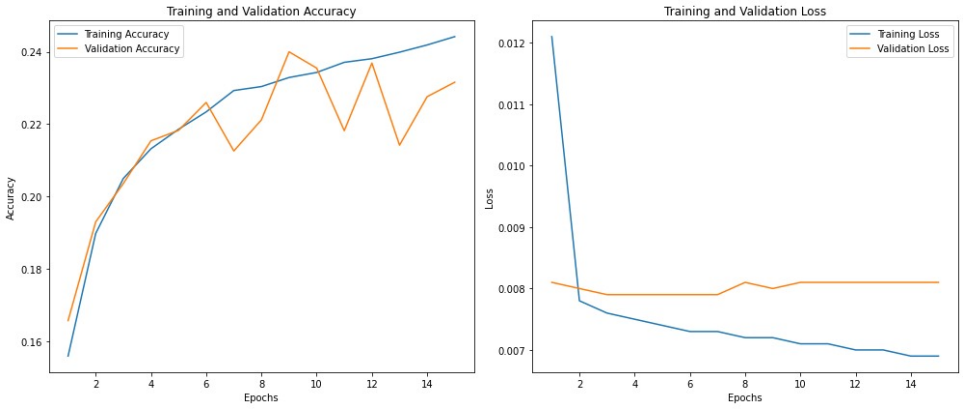

Figure 4: ResNet50 accuracy and loss graph.

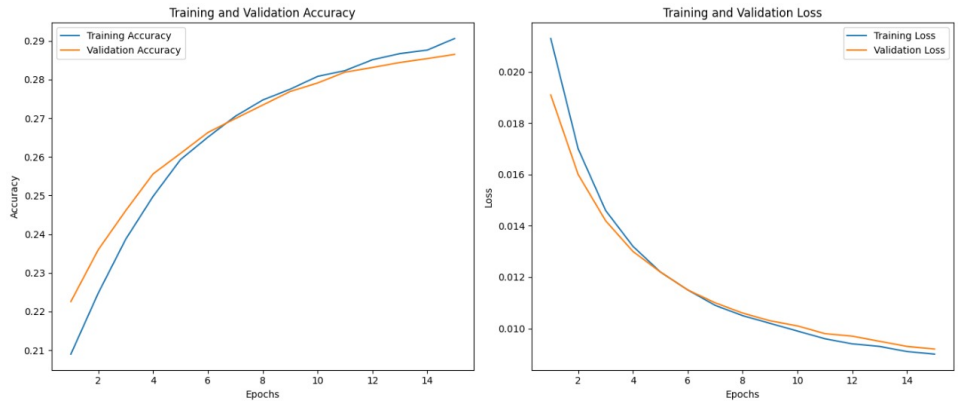

Figure 5: MobileNetV2 accuracy and loss graph.

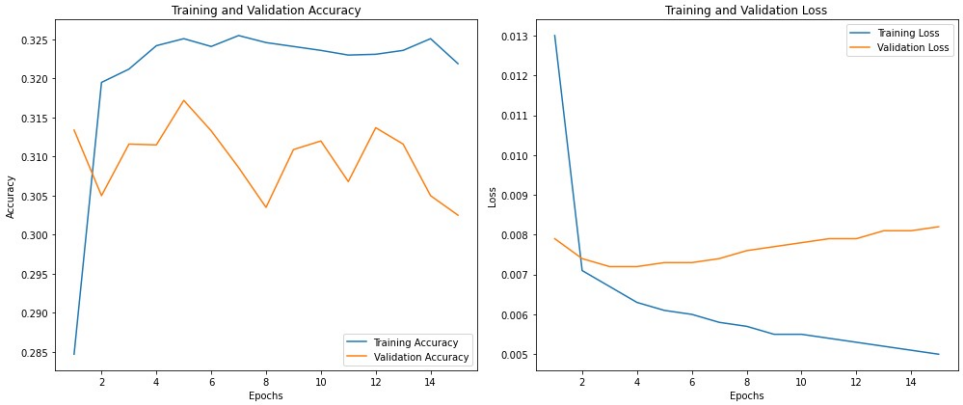

Figure 6: DenseNet121 accuracy and loss graph.

4. Conclusion

In this study, we addressed the Concept Detection Task of the ImageCLEFmedical Caption 2024 challenge, aiming to enhance automatic captioning and scene understanding of radiology images. We evaluated three deep learning models—ResNet50, MobileNetV2, and DenseNet-121—based on their ability to identify and locate relevant concepts within a large corpus of medical images. Among these models, ResNet50 demonstrated superior performance with an F1 score of 0.181, followed by MobileNetV2 with an F1 score of 0.178, and DenseNet-121 with an F1 score of 0.114. These results indicate that ResNet50 is the most effective model for concept detection in this context, providing the most accurate identification of individual components that form the basis for generating coherent captions. This work underscores the potential of using advanced convolutional neural networks in medical image analysis, contributing to the development of more efficient and reliable automated medical image captioning systems.

5. Acknowledgments

We would like to express our gratitude to Vellore Institute of Technology, Chennai for providing access to their advanced computational facilities. The models for this study were run on a Lenovo Thinkstation P348, which is equipped with an Intel Core i7-11700 processor @ 2.5 GHz (8 cores), 64 GB of RAM, a 2 TB hard disk, and a 12 GB NVIDIA graphics card. The robust hardware and high computational capabilities significantly contributed to the successful completion of this study.

References

[1] B. Ionescu, H. Müller, A. Drăgulinescu, J. Rückert, A. Ben Abacha, A. Garcıa Seco de Herrera, L. Bloch, R. Brüngel, A. Idrissi-Yaghir, H. Schäfer, C. S. Schmidt, T. M. G. Pakull, H. Damm, B. Bracke, C. M. Friedrich, A. Andrei, Y. Prokopchuk, D. Karpenka, A. Radzhabov, V. Kovalev, C. Macaire, D. Schwab, B. Lecouteux, E. Esperança-Rodier, W. Yim, Y. Fu, Z. Sun, M. Yetisgen, F. Xia, S. A. Hicks, M. A. Riegler, V. Thambawita, A. Storås, P. Halvorsen, M. Heinrich, J. Kiesel, M. Potthast, B. Stein, Overview of ImageCLEF 2024: Multimedia retrieval in medical applications, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction, Proceedings of the 15th International Conference of the CLEF Association (CLEF 2024), Springer Lecture Notes in Computer Science LNCS, Grenoble, France, 2024.

[2] J. Rückert, A. Ben Abacha, A. G. Seco de Herrera, L. Bloch, R. Brüngel, A. Idrissi-Yaghir, H. Schäfer, B. Bracke, H. Damm, T. M. G. Pakull, C. S. Schmidt, H. Müller, C. M. Friedrich, Overview of ImageCLEFmedical 2024 – Caption Prediction and Concept Detection, in: CLEF2024 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, Grenoble, France, 2024.

[3] O. Pelka, C. M. Friedrich, A. García Seco de Herrera, H. Müller, Overview of the ImageCLEFmed 2019 concept detection task, in: Proceedings of CLEF (Conference and Labs of the Evaluation Forum) 2019 Working Notes, 9-12 September 2019, 2019. URL: https://repository.essex.ac.uk/26557/.

[4] M. Rahman, A Cross Modal Deep Learning Based Approach for Caption Prediction and Concept Detection by CS Morgan State., in: CLEF (Working Notes), 2018, p. 8. URL: https://ceur-ws.org/Vol-2125/paper_138.pdf.

[5] K. Dimitris, K. Ergina, Concept detection on medical images using Deep Residual Learning Network, Working Notes CLEF (2017). URL: https://ceur-ws.org/Vol-1866/paper_122.pdf.

[6] L. Rossetto, M. Amiri Parian, R. Gasser, I. Giangreco, S. Heller, H. Schuldt, Deep learning-based concept detection in vitrivr, in: MultiMedia Modeling: 25th International Conference, MMM 2019, Thessaloniki, Greece, January 8–11, 2019, Proceedings, Part II 25, Springer, 2019, pp. 616–621. doi:10.1007/978-3-030-05716-9_55.

[7] K. Ohri, M. Kumar, Review on self-supervised image recognition using deep neural networks, Knowledge-Based Systems 224 (2021) 107090. URL: https://www.sciencedirect.com/science/article/pii/S0950705121003531. doi:10.1016/j.knosys.2021.107090.

[8] J. Rückert, L. Bloch, R. Brüngel, A. Idrissi-Yaghir, H. Schäfer, C. S. Schmidt, S. Koitka, O. Pelka, A. B. Abacha, A. G. S. de Herrera, H. Müller, P. A. Horn, F. Nensa, C. M. Friedrich, ROCOv2: Radiology Objects in COntext version 2, an updated multimodal image dataset, Scientific Data (2024). URL: https://arxiv.org/abs/2405.10004v1. doi:10.1038/s41597-024-03496-6.

[9] O. Pelka, S. Koitka, J. Rückert, F. Nensa, C. M. Friedrich, Radiology objects in context (roco): a multimodal image dataset, in: Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis: 7th Joint International Workshop, CVII-STENT 2018 and Third International Workshop, LABELS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings 3, Springer, 2018, pp. 180–189. doi:10.1007/978-3-030-01364-6_20.

[10] O. Bodenreider, The unified medical language system (UMLS): integrating biomedical terminology, Nucleic acids research 32 (2004) D267–D270. doi:10.1093/nar/gkh061.

[11] G. Huang, Z. Liu, L. van der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. doi:10.1109/CVPR.2017.243.

[12] Q. Ji, J. Huang, W. He, Y. Sun, Optimized deep convolutional neural networks for identification of macular diseases from optical coherence tomography images, Algorithms 12 (2019) 51. doi:10.3390/a12030051.

[13] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520. doi:10.1109/CVPR.2018.00474.

[14] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.