Active Learning for Pneumonia Detection

Martinus Boom
Group 31
uc2024154464@student.uc.pt

Abstract

Active learning (AL) has grown as an efficient machine learning approach, requiring considerably fewer labeled data to train models while maintaining performance. In this report, we evaluate the use of AL methods in a pneumonia detection system using convolutional neural networks (CNN). We aim to maintain or improve accuracy compared to the implemented passive learning baseline while significantly reducing the amount of labeled data required through the combination of an uncertainty sampling technique and iterative data labeling, and through that, we demonstrate how AL can effectively balance high performance with reduced data requirements in medical imaging applications.

1 Introduction

Medical imaging is increasingly using machine learning (ML) techniques to assist in diagnosis, particularly in the detection of diseases like pneumonia. Although ML models, particularly convolutional neural networks (CNN), achieve remarkable accuracy, their effectiveness depends on having access to a vast amount of labeled data. Getting such data in the medical industry requires extensive resources and professional annotations, which can be expensive and time-consuming. To achieve equivalent or better performance with significantly fewer labeled instances, Active Learning offers a potential solution to this issue by carefully choosing the most instructive samples for labeling.

This research project looks into the use of uncertainty sampling as a query approach in the application of AL to a pneumonia detection system. Unlike passive learning, which labels all data and uses it for training right away, AL iteratively searches the most questionable samples, gradually increasing model performance. Our goal is to significantly reduce the number of labeled samples needed while maintaining or exceeding the classification accuracy attained by a passive learning baseline.

Using CNN trained on a publicly available chest X-ray dataset, we apply an AL framework to identify cases of pneumonia. The model was first trained using the complete labeled dataset as a benchmark using a baseline passive learning technique. The AL approach was then used, with performance assessed at each iteration to compare accuracy and data efficiency. The objective of this project is to evaluate how AL can perform on the same level with or better than passive learning while requiring a much smaller number of labeled samples for pneumonia identification by utilizing uncertainty sampling and iterative data labeling.

The resources and techniques used, including the established AL framework and its relationship to CNN for pneumonia diagnosis, are covered in the chapters that follow. Also then discuss the results, which evaluate how far AL reduces the requirement for labeled data while maintaining performance, and end with possible future paths for this methodology.

2 Related Work

Active learning capacity to lower labeling while preserving high accuracy has made it popular in medical image analysis. Studies have demonstrated the efficacy of uncertainty sampling strategies, such as least confidence and entropy-based algorithms, in prioritizing useful samples for medical datasets [Settles, 2009], especially useful for chest X-ray analysis, where interpretations require specialist knowledge. Despite these benefits, many studies ignore the problem of class imbalance, which may reduce model performance, so this project resolves this issue by applying the Synthetic Minority Oversampling Technique (SMOTE) to the training data used in the active learning process.

CNN architectures frequently used for pneumonia identification, such as those benchmarked in the Kaggle Chest X-ray Pneumonia dataset [Kermany et al., 2018], are the foundation for baseline passive learning in this work, such thing with data augmentation techniques like flipping, rotation, and brightness change are congruent with existing methodologies for improving generalization in passive learning models. On the other way, these techniques are not successful when labeling resources are small, as they require completely labeled datasets.

As previously mentioned, addressing class imbalance is a key component of medical image analysis, and this project builds on SMOTE by maintaining balance in both the initially labeled dataset and the queried samples during active learning, overcoming challenges in detecting diabetic retinopathy and breast cancer, as previously reported [Chawla et al., 2002].

This type of study highlights the advantages and disadvantages between both active and passive learning, while passive learning provides a consistent performance with large labeled datasets [Kermany et al., 2018], active learning improves labeling efficiency by focusing on informative samples [Settles, 2009], additionally, this project shows that, with a balanced dataset, active learning may match passive learning's accuracy while using much fewer labeled samples.

3 Materials an Methods

In order to maximize the labeling process, this methodology starts with a baseline passive learning model and applies active learning. Only the most informative samples are iteratively labeled in uncertainty sampling, which is based on a convolutional neural network (CNN). These steps aim to achieve competitive performance while considerably lowering labeled dataset size.

3.1 Baseline Passive Learning Model

The baseline model used in this project was based on an available version of passive learning for pneumonia diagnosis using convolutional neural networks. This method was taken from a Kaggle notebook[] that classified chest X-ray images into two categories: "Pneumonia" and "Normal."

The passive learning method includes training a model on a fixed dataset with no iterative labeling or feedback methods. This simple procedure served as the baseline for assessing the efficacy and performance of active learning in this study.

Key aspects of the passive learning implementation:

- **CNN Architecture:** The model used a deep CNN that could extract hierarchical characteristics from grayscale X-ray pictures. The architecture was designed for binary classification and achieved a test accuracy of 92.6%.
- **Data Augmentation:** To combat overfitting and increase generalization, the model used an augmentation process. Random rotations, magnification, horizontal flips, and tiny translations were implemented to expand and diversify the datasets.
- **Performance on the Test Set:** The test accuracy of the passive learning model served as an essential benchmark for the stopping condition in active learning, guaranteeing that iterations were only maintained when validation accuracy matched or exceeded the baseline, reducing labeling effort while preserving performance.

The passive learning implementation served as a reliable baseline against which the efficacy of active learning was measured. The goal was to see if active learning could achieve equivalent or higher accuracy while drastically lowering the number of labeled samples necessary.

3.2 Dataset, Preprocessing, and Augmentation

The dataset used in this project was taken from Kaggle and includes chest X-ray images classed as "Pneumonia" and "Normal."

It was divided into three subsets: training, validation, and test. The training set was used to train the passive learning model and contained the labeled and unlabeled pools for the active learning framework. The validation set evaluates intermediate performance during training and active learning iterations, assisting in model optimization and lead-stopping criteria. The test is for final evaluation to guarantee an equal evaluation of the model generalization ability.

Preprocessing Steps

To ensure consistency and increase efficiency, all subsets went through the following preprocessing steps:

- **Resizing:** All photographs were downsized to 150x150 pixels to standardize input dimensions and reduce computational complexity.
- **Grayscale Conversion:** Images were converted to grayscale, which removed unnecessary color information while keeping diagnostic features.
- **Normalization:** Pixel values were scaled to the range [0, 1], which resulted in more consistent input distributions and better model convergence.

The dataset splitting and preprocessing methods originated from the Kaggle passive learning implementation.

Data Augmentation

Created to improve model generalization and reduce overfitting, a data augmentation pipeline based on the same Kaggle implementation as the baseline passive learning model dynamically generates versions of the training data by applying the following transformations:

- **Rotation:** Random rotations by 30 degrees.
- **Zoom:** Random zooms by 20%.
- **Width and Height Shifts:** Random translations by 10% of image dimensions.
- **Horizontal Flipping:** Random flipping images horizontally.

This augmentation strategy was incorporated into the active learning framework to guarantee consistency and equality of comparison. It enhanced active learning's iterative labeling process by adding variability to the classified dataset. This diversity reduced overfitting and exposed the model to more data patterns, boosting its capacity to generalize effectively to previously unseen samples.

3.3 Data Balancing

Once create the dataset we notice as pic shown in Figure 1, there is a imbalance of class in the name of "Pneumonia", which has more sample than "Normal", it will impact the model prediction as it favoring the majority class and result unfavorable result for "Normal" category.

Data Augmentation Limitations

Data augmentation approaches were used in order to increase the diversity of the training dataset, however, it does not address the class imbalance problem. Augmentation gives rise to variations of existing samples of the class but does not provide extra minority class samples required to make the data set balanced.

SMOTE

To balance this imbalance , the Synthetic Minority Oversampling Technique (SMOTE) was utilized, creating new samples for the minority class by combining existing samples. This maintains variance and mitigates the risk of overfitting to duplicate samples.

Application in Active Learning

In SMOTE application, **x_train**, the entire training data has been used to implement the active learning which focuses on ensuring that both the initial labeled and the unlabeled pools are drawn from a dataset that is balanced. This made it possible for the model to better characterize the under represented group “Normal” and increase the chances of effectively querying. This is in sharp contrast with the passive learning process that applied data augmentation techniques, where SMOTE was applied for active learning to solve the class imbalance problem for target classes.

```
# Apply SMOTE to the entire training dataset
n_samples, width, height, channels = x_train.shape
x_train_flat = x_train.reshape(n_samples, -1)

smote = SMOTE(random_state=42)
x_train_resampled_flat, y_train_resampled = smote.fit_resample(x_train_flat, y_train)

# Reshape the resampled data back to the original shape
x_train_resampled = x_train_resampled_flat.reshape(-1, width, height, channels)
```

For the usage of SMOTE, the training dataset was first reduced to a single dimension and to match the requirements of the CNN model, the dataset was returned to its original structure after creating synthetic samples.

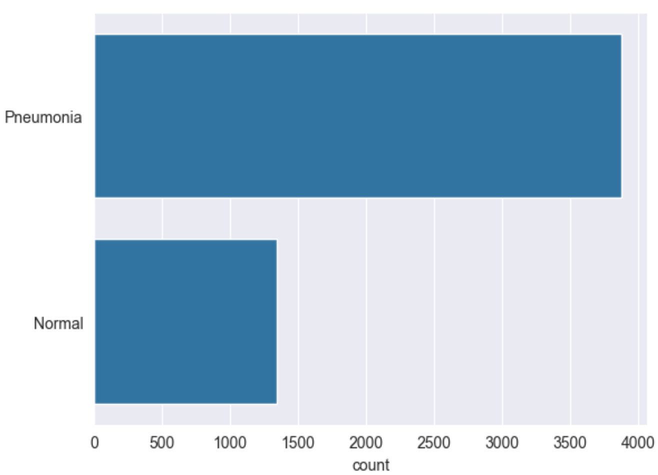

Figure 1: Class distribution in the dataset.

3.4 Active Learning Framework

The active learning framework was implemented to iteratively query the most informative samples from an unlabeled pool, add them to the labeled set, and then retrain the model. This method is aimed to reduce the number of labeled samples needed to obtain comparable accuracy to the passive baseline model.

Query Strategy

Uncertainty sampling was used as the query approach because it detects unknown information where the model is less confident. This method chooses data points with predicted probabilities closest to the decision boundary, making it appropriate for binary classification applications. The project’s objectives are in keeping with the computational efficiency and direct targeting of informative situations of uncertainty sampling, as opposed to classification margin, which concentrates on the smallest difference between the top probabilities, and classification entropy, which assesses total uncertainty.

```
def custom_uncertainty_sampling(learner, X_pool, n_instances=10):
    probas = learner.estimator.predict(X_pool)  # Predict probabilities for the samples in the pool
    uncertainty = np.abs(probas - 0.5).flatten()  # Highest uncertainty is closest to 0.5
    query_idx = np.argsort(uncertainty)[:n_instances]
    return query_idx, X_pool[query_idx]
```

In order to bridge the gap between the modAL library and Keras models, the **custom function** was created to replicate the behavior of the uncertainty sampling methods that are already included in the modAL. Because Keras does not natively support modAL’s **predict_proba** method, this function calculates uncertainty, ensuring compatibility while maintaining functionality.

Iterative Labeling

The active learning framework’s iterative labeling procedure starts with a small, labeled subset (5% of the training data) that is chosen to kick off the active learning cycle. This allows the model to incrementally improve its performance by focusing on the most informative data during each iteration.

At each iteration, are performed the following steps:

1. **Querying Samples:** Using the uncertainty sampling method, the model selects ten unlabeled samples from the pool that are calculated to be the most informative.
2. **Labeling:** The query samples are manually identified and moved from the unlabeled pool to the labeled dataset.
3. **Updating the Dataset:** The newly labeled samples are included in the labeled dataset providing the model with more diverse and useful data.
4. **Retraining the Model:** For better classification limits, the model is retrained with the new labeled dataset, which now includes more data.

Starting with a small labeled set and gradually expanding it, reducing the quantity of labeled data required while improving or maintaining the accuracy of the baseline passive learning model.

Model Retraining

The model was retrained after each iteration with the updated labeled dataset. To artificially expand the quantity and variety of the labeled data, the data augmentation process from the baseline model was kept, and the learner.teach() method from the modAL package was utilized for retraining.

Since the teach() method in modAL does not support Keras callbacks such as **ReduceLROnPlateau**, a manual learning rate reduction was implemented to achieve consistent improvement by monitoring validation accuracy and changing the learning rate when performance decreased, replicating the behavior of the passive learning model.

```
# Manual learning rate reduction logic
    if val_accuracy_iter > best_val_accuracy:
        best_val_accuracy = val_accuracy_iter
        wait = 0  # Reset patience if
            performance improves
    else:
        wait += 1
        if wait >= patience:
            current_lr = learner.estimator.
                optimizer.learning_rate.numpy
                ()
            new_lr = max(current_lr * factor,
                 min_lr)
            learner.estimator.optimizer.
                learning_rate = new_lr
            print(f"Reducing learning rate to
                 {new_lr:.6f}")
            wait = 0  # Reset patience after
                reducing learning rate
```

3.5 Stopping Criterion

Using a tolerance (α) of 2.5%, the stopping condition for the active learning process was determined. The iterations were stopped when the active learning model's validation accuracy reached the same level as the passive learning model's test accuracy. This adaptable strategy ensures that active learning meets its main goal, efficient use of labeled data while maintaining excellent performance.

```
# Stopping Condition
    if val_accuracy_iter >=
        test_accuracy_passive - alpha:
        print("Stopping criteria met: Active
            Learning accuracy is within the
            defined threshold.")
        break
```

3.6 Evaluation Metrics

Equal to the passive baseline, accuracy, and loss were selected as the main criteria to assess the active learning framework performance. These metrics were produced at each iteration for both the training and validation datasets using Keras **.evaluate()** method, ensuring a consistent evaluation of the model learning from augmented labeled data and generalization to unseen samples.

1. **Accuracy:** The proportion of right predictions is measured, providing a clear benchmark for determining whether the framework performed as well as or better than the baseline.
2. **Loss:** Determined how successfully the model reduced inequalities during training by quantifying the difference between expected and actual values.

To take care of the lack of validation monitoring in the Keras **.teach()** method, metrics were tracked across iterations using lists (**train_acc, train_loss, val_acc, val_loss**), capturing model learning progress and its capacity using validation metrics. By determining the stop state based on the tolerance threshold (α), these metrics made it possible to compare accuracy and loss trends, showing how active learning reduced labeling effort while preserving or improving baseline performance.

4 Results and Discussion

4.1 Results

Accuracy and loss measures showed only minor variations between the active learning framework and the baseline passive learning model. Both methods final test results are displayed in Table 1.

Metric	**Passive Learning**	**Active Learning**
Test Accuracy (%)	90.87	91.35
Test Loss	0.275	1.222

Table 1: Test Performance Metrics for Passive and Active Learning Models

Performance Across Iterations

Figures 2 and 3 represents the training and validation accuracy and loss patterns for the active learning framework across iterations, and Figures 4 and 5 shows similar metrics results for the passive learning model across epochs.

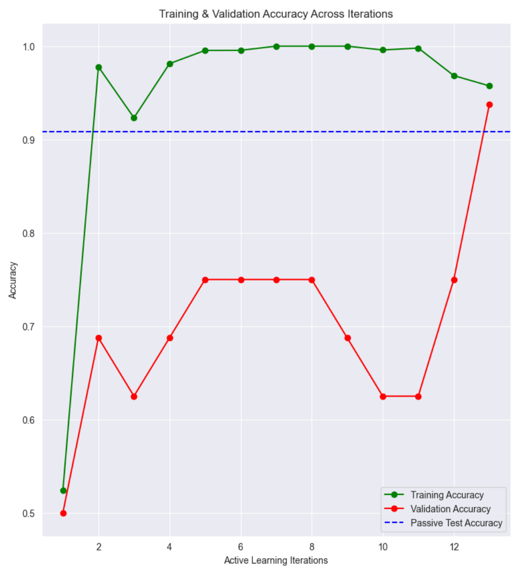

Figure 2: Training and validation accuracy for Active Learning.

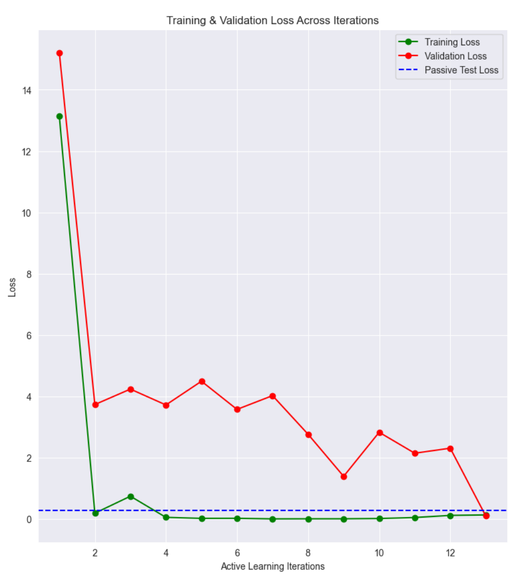

Figure 3: Training and validation loss for Active Learning.

Active Learning: The validation process accuracy increased over time until it reached that of the passive learning test, and until the labeled pool expanded, early iterations revealed generalization difficulties due to oscillations in validation loss caused by the smaller labeled dataset.

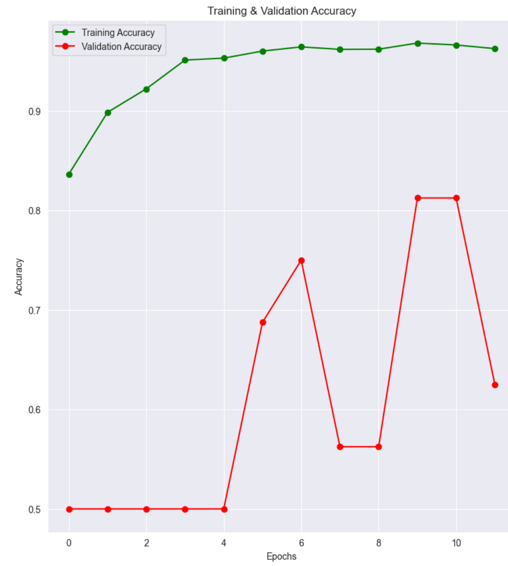

Figure 4: Training and validation accuracy for Passive Learning.

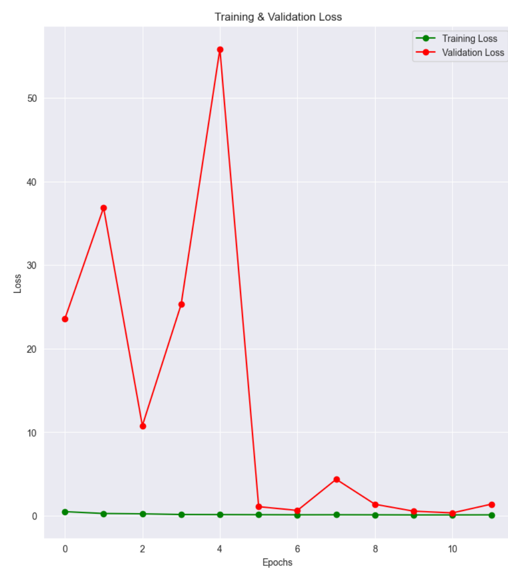

Figure 5: Training and validation loss for Passive Learning.

Passive Learning: It demonstrated a consistent reduction in loss and gains in accuracy, having a completely labeled dataset, for faster training and stronger generalization.

Confusion Matrix Analysis

Figures 6 and 7 show the confusion matrices for active and passive learning models, respectively. Both techniques performed well in recognizing pneumonia cases but struggled in the minority class ("Normal").

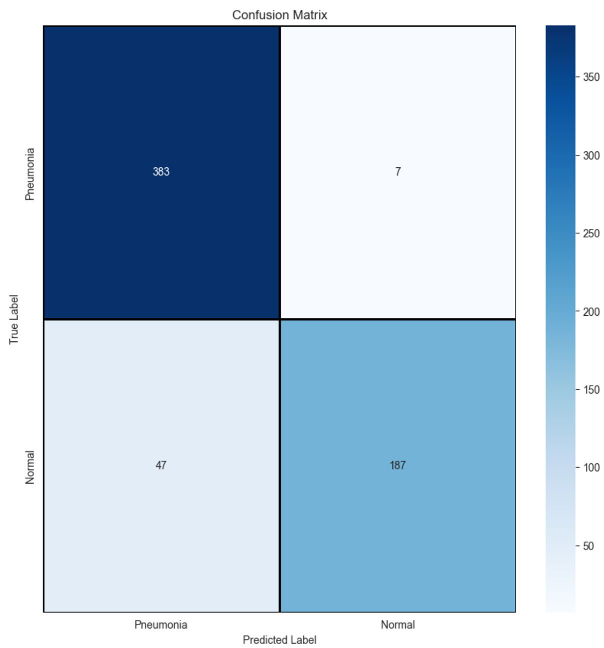

Figure 6: Confusion Matrix of Active Learning Model.

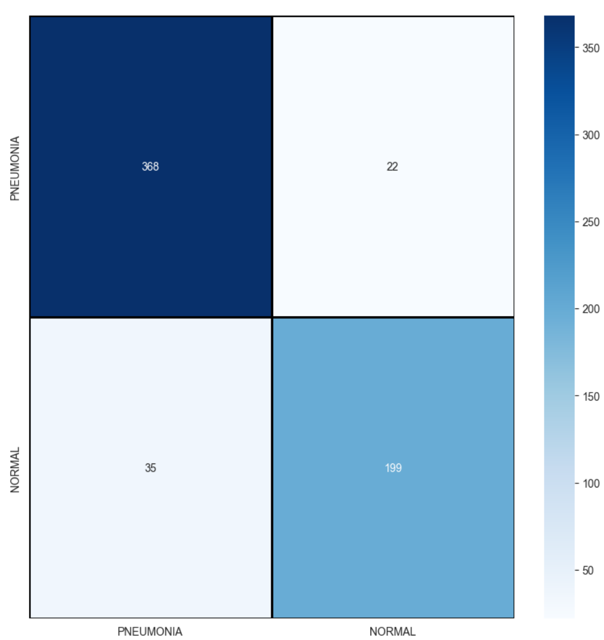

Figure 7: Confusion Matrix of Passive Learning Model.

4.2 Discussion

In the context of medical picture classification, the findings highlight the benefits and limitations of active and passive methods of learning. Active learning demonstrated its ability to improve labeling efforts compared to passive learning with way more labeled examples and using the stopping condition, the framework required only 500 labeled images, beginning with 5% of the balanced dataset generated by SMOTE and adding just 13 sets of 10 samples each. This efficiency stands in contrast to the 5,216 training images utilized in passive learning, demonstrating the promise of active learning under conditions where labeling data is limited.

However, active learning had some early challenges, as indicated by various validation losses during the first iterations, and as the labeled dataset grew, the model validation accuracy improved, meeting the passive learning baseline and overcoming early difficulties, showing the importance of active learning in identifying and using the most useful data.

Passive learning, on the other hand, produced a more constant trajectory of accuracy improvement and loss reduction during the training due to its dependence on a fully labeled dataset from the start, which allows for more robust generalization and smoother resolution. Even so, the approach's dependence on a large labeled dataset makes it less efficient in real-world circumstances where labeling costs are limitations. In addition, passive learning did not address the dataset's class imbalance, focusing instead on data augmentation to improve diversity and generalization.

The observed differences in test accuracy and loss metrics between the two learning frameworks provide useful information about their behavior. While active learning had a little higher test accuracy (91.35%) than passive learning (90.87%), the resulting test loss was considerably greater (1.222 vs. 0.275), which indicates that, while active learning was good at making correct predictions, it may have faced more inconsistency in some cases, possibly due to its iterative labeling process and smaller labeled dataset. In contrast, lower loss reflects the capacity to consistently minimize errors, which is the result of having access to a fully labeled training dataset.

Overall, the results indicate that active learning can be an efficient alternative to passive learning, especially in the lack of data or cost fields such as medical imaging. For purposes requiring high-quality predictions with small labeling effort, the framework offers a solution by combining efficiency and performance through the integration of methods like SMOTE and uncertainty sampling.

5 Conclusion

To summarize, the goal of this project was to evaluate active and passive learning approaches in terms of efficiency and performance in medical image classification, with a focus on minimal labeled data. The results revealed that active learning, using iterative sampling strategies such as uncertainty sampling and data balancing methods such as SMOTE, achieved comparable test accuracy to passive learning while significantly reducing labeling, and using the stopping criterion, the active learning framework required only about 500 labeled images, compared to 5,216 training images in the passive baseline, which demonstrates that active learning can be a viable option in situations where resources are limited.

Although its efficiency, active learning had early-stage issues, most notably changing validation loss during the first iterations, which placed the model dependence on a limited initial labeled dataset. Future work could solve these early-stage limitations by improving generalization and stability during the first iterations, and additionally, hybrid techniques that combine the robustness of passive learning with the efficiency of active learning, showing promise of improving model performance in real-world medical applications.

References

- Boom, M. (2024). Active Learning for Pneumonia Detection. GitHub repository containing the implementation of the project. https://github.com/MartinusBoomUc2024154464/AL_PneumoniaDetection/tree/main.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321–357. https://www.jair.org/index.php/jair/article/view/10302.
- Kermany, D. S., Goldbaum, M., Cai, W., et al. (2018). Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell*, 172(5), 1122–1131.e9. https://doi.org/10.1016/j.cell.2018.02.010.
- Settles, B. (2009). Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin-Madison. https://pages.cs.wisc.edu/~bsettles/pub/settles.activelearning.pdf.
- Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2017). Understanding Deep Learning Requires Rethinking Generalization. *arXiv preprint arXiv:1611.03530*. http://arxiv.org/abs/1611.03530.
- Sener, O., & Savarese, S. (2018). Active Learning for Convolutional Neural Networks: A Core-Set Approach. *arXiv preprint arXiv:1708.00489*. http://arxiv.org/abs/1708.00489.
- Kaggle. (n.d.). Chest X-Ray Images (Pneumonia). Dataset. https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia.
- Kaggle. (n.d.). Pneumonia Detection using CNN (92.6% Accuracy). https://kaggle.com/code/madz2000/pneumonia-detection-using-cnn-92-6-accuracy.
- modAL Documentation. (n.d.). ActiveLearner. https://modal-python.readthedocs.io/en/latest/content/models/ActiveLearner.html.
- modAL Documentation. (n.d.). Uncertainty Sampling. https://modal-python.readthedocs.io/en/latest/content/query_strategies/uncertainty_sampling.html.
- modAL Documentation. (n.d.). Keras Models in modAL Workflows. https://modal-python.readthedocs.io/en/latest/content/examples/Keras_integration.html.
- Dey, A. (2024). A Gentle Introduction to Active Learning. *Medium*. https://arindam-dey.medium.com/a-gentle-introduction-to-active-learning-e983b9d175cb.
- Kaggle. (n.d.). Accuracy vs Loss Conflict. https://www.kaggle.com/discussions/general/a.
- Deep Learning Book. (n.d.). https://www.deeplearningbook.org/.