National College of Ireland

A novel AutoML library for pre-defined transfer learning workflows

MSc Research Project
Data Analytics

Vivek Vindhyachal Rai
Student ID: x21142157

School of Computing
National College of Ireland

Supervisor: Dr Giovani Estrada

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Vivek Vindhyachal Rai
Student ID:	x21142157
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Dr Giovani Estrada
Submission Due Date:	09/09/2023
Project Title:	A novel AutoML library for pre-defined transfer learning workflows
Word Count:	6928
Page Count:	17

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	4th December 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A novel AutoML library for pre-defined transfer learning workflows

Vivek Vindhyachal Rai
x21142157

Abstract

The field of artificial intelligence has experienced significant advancement due to the rapid growth of machine learning & pre-defined transfer learning techniques. As machine learning models become more complex there is a growing demand for automated and user-friendly model development. This research introduces an AutoML library that simplifies the utilization of pre-defined transfer learning workflows. The research includes the AutoML software which works with various pre-defined transfer learning models such as VGG19, ResNet-50, Imagenet, DenseNet-121 and EfficientNet-B7. These options are well-known for their versatility and outstanding results. We will show how users can bring their image classification datasets and use the our automl developed library to receive a comprehensive output. It includes performance measures like accuracy, precision, recall and F1 score for a wide range of transfer learning algorithms.To evaluate the new library two image classification use cases are presented with encouraging results. By addressing this research gap it is expected AutoML to play a greater role in simplifying and ensuring equal opportunities in machine learning empowering a wider audience and promoting further progress in the automated development of models.

1 Introduction

The arrival of automated machine learning (AutoML) has facilitated the democratisation of machine learning algorithms by automating pre-processing steps as well as model generation and selection steps. The use of pre-defined transfer learning approaches in the field of Automated Machine Learning (AutoML) has also gained popularity due to its capacity to leverage pre-existing models hence minimising the need of training models from the ground up Salehin et al. (2023).

The AutoML collection includes a diverse range of pre-trained deep learning models including VGG19 Simonyan and Zisserman (2015) , ResNet-50 He et al. (2015), Imagenet Howard et al. (2017), DenseNet-121 Huang et al. (2018) and EfficientNetB7 Tan and Le (2020). The selection of these models has been based on their established reputation and versatility across many jobs and shown effectiveness. With the increasing growth of machine learning applications there arises a need for adaptable and effective tools capable of addressing a wide range of challenges. The incorporation of several approaches inside the library enhances its range and applicability across many sectors and research domains.

The two areas AutoML and transfer learning are the research focus of this report. We investigates the design and evaluation of an AutoML library. The text provides an analysis of the architectural factors, constituent aspects and methodologies related to transfer learning. The trials provide valuable insights on the tangible effects of the library and the advantages and drawbacks of each used approach.

In the current period of machine learning the essential factors that drive progress and success are automation and accessibility. The creation of an AutoML library dedicated to transfer learning is a notable advancement in the field. The primary objective of this thesis is to run different transfer learning model and help user to easily select the best model based on there performance therefore providing researchers and practitioners in the field of machine learning with enhanced capabilities. This study aims to facilitate the democratisation of machine learning and serve as a source of inspiration for future advancements in automated model creation.

The subsequent chapters will describe the development process of the AutoML library the performed tests aimed at evaluating its capabilities and a comprehensive examination of the obtained findings. This elucidates the comprehensive comprehension of the significance that this work offers onto the machine learning community.

1.1 Research Gap

To our best knowledge there is no AutoML libraries for transfer learning. Therefore the primary objective of this thesis is about the development of an AutoML library designed primarily for the purpose of transfer learning. The library should provide users with an easy way to apply well-established transfer learning algorithms given a classification dataset. Individuals submit their dataset and are provided with a thorough table that displays performance numbers for each strategy. The primary objective of the library is to implement automated processes and enhance efficiency in the tasks of model selection and assessment catering to individuals with varying levels of expertise in the field of machine learning.

1.2 Research question

In the scope of the current research report, two questions came to the fore:

RQ1 To what extent could transfer learning tasks be automated?

RQ2 How do we design a library for the automation of transfer learning workflows?

The first question will deal with intricacies of pre-processing and model preparation for transfer learning. It is not clear to what extent these steps can be automated across a variety of transfer learning models. While the second question will focus on the design trade-offs to make a library work, conceptually similar to PyCaret[1], but for transfer learning workflows.

1.3 Research objectives

The research presented here required a number of steps:

1. Understand how transfer learning and AutoML concepts work. In particular the focus is to design a library akin to PyCaret.

[1] `https://pycaret.org/`

2. Implement different transfer learning models available in Keras.
3. Understanding Data agumentation and using it in our development of AutoML Library.
4. Develop our AutoML library.

1.4 Outline

Finally, we can now close Section 1 by outlining the structure of the report, for instance: The remainder of the report is organised as follows. Section 2 presents the relevant theory and works closely related to the proposed one. Section 3 describes details of the proposed approach. In Section 4, we describe the techniques that are used to address the problem, as well as all proposed test cases. The two novel methods are presented in Section 5, while evaluation results appear in Section 6. Finally, we provide conclusions and discussions of future research directions in Section 7.

2 Related Work

A literature review is necessary to complete a research project. We also want to provide context for the use cases, as to show what classification accuracy is expected when transfer learning or deep learning is applied manually. It will thus provide guidance on what level of accuracy is good enough for the newly developed AutoML library.

This section will critically analyze previous research on AutoML and transfer learning, as well as use cases for emotion and pneumonia detection. Conducting a thorough literature review is important for building upon existing work and establishing a guidance for our current research.

2.1 Work done in automated machine learning area

The approach proposed by He et al. (2021) uses meta-feature extraction and a dynamic dataset clustering algorithm to reuse models for multiple datasets, reducing search time. It can be combined with other AutoML techniques like genetic methods, reinforcement learning, Hyperband, and DARTS.

The proposed approach is compared with various baselines including traditional statistical and categorised meta-feature generating methods, random grouping, k-means and multi-task solutions for Bayesian optimisation. An promising AutoML approach that reduces search time without compromising accuracy is proposed by the author Xue et al. (2019). It has been indicated by experimental results of image classification for multiple datasets that the proposed approach performs better than the baselines in terms of accuracy and search time.The flexibility of the proposed approach is discussed in the paper including its ability to handle different search algorithms and datasets under a limited time budget. The online setting is also applicable.

The possibility of creating deep learning systems without human assistance and interference is being explored by AutoML. An emerging field in machine learning. A comprehensive review on this topic can be found in a recent paper by author He et al. (2021). In this paper we will explore the applications and limitations of deep learning, with a specific focus on AutoML as a potential solution. We examine and compare various techniques with a particular focus on neural architecture search (NAS) using popular datasets. Open problems in AutoML are also identified by the authors who propose future research directions. These directions include the development of efficient NAS algorithms and the addressing of resource-aware NAS challenges with the aim of improving scalability and generalisation of AutoML methods.

A recent paper Doke and Gaikwad (2021) reviews current research on Automated Machine Learning and Meta Learning, emphasizing the need to automate machine learning for cost and time efficiency. It also examines challenges faced by data scientists dealing with large amounts of data, compares AutoML with traditional ML, and discusses available datasets and tools. The conclusion summarizes current trends in AutoML and Meta Learning.

A new group of domain-specific meta-features for selecting models in anomaly detection tasks is suggested in the paper Kotlar et al. (2021). The paper examines current meta-learning approaches and discusses difficulties involved in creating a set of meta-features that accurately represent the characteristics of anomalies within data. A set of domain-specific meta-features for anomaly detection is defined, and an architecture design for the model selection system used in experiments is presented. The efficacy of the proposed set of meta-features is evaluated through experiments that answer research questions. In conclusion, the paper summarizes advantages and disadvantages associated with the proposed set of meta-features and proposes new areas for future research.

2.2 Work done on transfer learning

A new method for detecting emotions in animated characters was proposed using deep and transfer learning techniques Ghosh et al. (2022). Facial features were extracted with Uniform Local Binary Patterns (LBP) and fed to a pre-existing Convolutional Neural Network (CNN) model based on ResNet50 and ResNet101. The CNN was trained with 1984 stylized character images to classify seven emotions. Models were evaluated based on accuracy, precision, recall, and F1-score. ResNet50 and ResNet101 had accuracies of 98.7% and 99.8%. The proposed approach outperformed existing models in identifying emotions for animated faces The paper discusses limitations, such as small data and need for further research. However, it proposes a promising method for detecting emotional expressions in animated characters through deep learning and transfer learning, which could have applications beyond the animation industry.

Transfer learning techniques like subspace alignment, transfer component analysis, and deep learning-based methods. Transfer learning for visual categorization has also been studied by Shao et al. (2015) These techniques adapt the feature representations of the source domain to the target domain, enabling the target model to learn from the source domain's knowledge.The paper discusses classifier-level knowledge transfer methods like instance transfer, domain adaptation, and multi-task learning. These techniques aim to transfer knowledge from the source domain's classifier to the target domain's classifier, helping the target model learn from the source domain's classification experience. The paper discusses model selection methods such as multiple kernel learning and cross-validation. These methods help choose the most suitable transfer learning technique for a visual categorization task, considering the characteristics of the source and target domains. The authors analyze different techniques, discussing their pros and cons. They evaluate and compare the performance of various transfer learning techniques in different visual categorization tasks.

2.3 Work done in facial emotion detection

Emotion detection is one of the selected use cases. Therefore, it makes sense to briefly review some work done on this field. Some authors have proposed CNNs and OpenCV to detect and classify emotions like neutral, happy, sad, surprise, angry, fear, and disgust from frontal facial expressions captured by a live webcam Giri et al. (2022). The authors review previous studies on emotion detection through facial recognition. They discuss potential applications in human-computer interaction, such as improving virtual assistants and chatbots. The paper outlines a methodology for implementing an emotion detection algorithm, covering facial image preprocessing, CNN-based feature extraction, and SVM-based emotion classification. The authors demonstrate the effectiveness of their approach with 92.5 percent accuracy on a set of 1,080 facial images. The paper offers a valuable approach to emotion detection through facial recognition. It has potential applications in various fields, and achieves high accuracy. The authors review literature, provide a detailed methodology, and present experimental results that support their approach's effectiveness.

Another work introduced a deep learning-based method for facial emotion detection using convolutional neural network architecture Jaiswal et al. (2020). It evaluates the method on two datasets, FERC-2013 and JAFFE, with high accuracies. It also includes a brief literature review of previous works in facial expression analysis and deep learning-based image classification. This contributes to the field of facial emotion detection and demonstrates the effectiveness of deep learning methods with an average accuracy of 70.14% on FERC-2013 dataset and 98.65% on JAFFE dataset, higher than previous models. The paper also presents a confusion matrix of classification accuracy for FERC-2013 dataset.

A hybrid object detection model for facial expression recognition in elderly mental health care was proposed recently Khajontantichaikun et al. (2022). It achieves an accuracy of 94.07% using an ensemble voting method with three different models. The significance, limitations, and ethical concerns of using this technology are also discussed. The study was supported by the Virtual Companion and Monitoring System for the Elderly project under the 2019 National Research Council of Thailand (NRCT) grant in Thailand.

An enhanced Yolov5s algorithm for emotion detection has also been proposed by author Zhang et al. (2022). By combining neighbouring cell prediction boxes it employs a merge technique to increase prediction box accuracy. The authors uses a dataset of womens facial expressions to evaluate the performance of the enhanced model by author. In this paper author enchanced approach shows that it had higher scores and improved detection accuracy when compared to the original Yolov5 model. The paper makes a significant addition to the field of facial expression recognition research and emphasizes the significance of deep learning-based target identification methods.

Automatic emotion change detection through facial expression analysis i this paper the authors of this paper Han et al. (2023) identify significant changes in facial expressions in films without the need for expensive temporal annotations by using a weakly-supervised deep learning system. On three datasets their technique achieves state-of-the-art performance. They also adapted it for temporal spotting with similar outcomes.The limitations of the strategy in identifying micro-expressions are emphasised throughout the study. The authors suggest further study in the future to enhance the framework. They also talk about the possible uses of this technology such as mental health monitoring systems and virtual assistants that can recognise emotions. The research offers a thorough investigation of deep emotion change detection by facial expression analysis, proving the efficacy of the suggested approach. Three datasets were used to assess the accuracy of the proposed method: CASME II, MMI, and YoutubeECD. On CASME II, it outperformed the state-of-the-art approach by 2.5% with an accuracy of 87.5%. With an accuracy of 85.7% on MMI, it outperformed the state-of-the-art approach by 1.7%. With an accuracy of 83.3% on YoutubeECD, it outperformed the state-of-the-art approach by 3.3%. Using cutting-edge techniques on CASME II the authors also adjusted their framework for temporal spotting.

2.4 Work done in pneumonia detection

Transfer Learning and Convolutional Neural Networks (CNN) have been used in the detection pneumonia from chest X-rays Kalgutkar et al. (2021). The authors highlighted the significance of early diagnosis and the global impact of pneumonia, which leads to over 2.7 million deaths. The study aims to address issues with inaccurate X-ray readings and the need for a precise, fast, and universal diagnostic solution. The study examines how CNN models like InceptionV3, ResNet-50 and VGG16 can be used for pneumonia detection through Transfer Learning. Transfer Learning involves using pre-trained models as a foundation for new applications. This approach in deep learning saves computing power, time, and resources typically needed to train layers for image classification problems. The study improves accuracy by 5-7% with customized neural layers. These layers and data augmentation prevent overfitting in transfer learning models with limited data. The study applies the No Free Lunch Theorem to build and compare various models, as no single algorithm works best for all problems. The paper presents evaluation metrics used to test model efficiency, including Binary Accuracy, AUC, Recall, Precision, and Confusion Matrix. The authors conclude that AUC is a better metric for evaluating model performance. Additionally, it presents evaluation metrics concluding that AUC is a better metric for assessing model performance.

Table 1: Tabular summaries of literature reviews

Paper	Description	Drawbacks
He et al. (2021)	a general overview of AutoML and its advantages and limitation;	no actual implementation
Doke and Gaikwad (2021)	introduction to AutoML and meta learning;	no critical analysis of the limitations and challenges
Kotlar et al. (2021)	domain-specific meta-features for automated machine learning;	limited to a relatively small number of datasets.
Giri et al. (2022)	facial recognition using CNNs and OpenCV with 92.5% accuracy ;	limited sample size and potential biases in the dataset
Jaiswal et al. (2020)	Emotion detection using Deep learning with 70.14 to 98.65 accuracy	does not discuss the limitations of the proposed method
Khajontantichaikun et al. (2022)	a hybrid object detection model for elderly facial expression 94.07% accuracy	potential limitations and ethical concerns of using facial data
Zhang et al. (2022)	improved Yolov5s for real-time emotion detection in online education	dataset used for evaluation only contains women data
Han et al. (2023)	deep learning framework for automatic emotion change detection with 87.5% to 85.7% accuracy	limitated in terms of detecting micro-expressions
Ghosh et al. (2022)	novel approach for facial emotion detection in animated characters 50:98.7% to 99.8% accuracy	proposed approach was evaluated on a small dataset
Yuvaraj et al. (2023)	comparative study of machine learning methods for human face emotion recognition	only limitation it considers two image datasets
Shao et al. (2015)	various transfer learning categories, including feature representation and classifier transfer	less detailed comparison of different transfer learning
Xue et al. (2019)	Transferable AutoML by Model Sharing over Grouped Datasets with 72.5% to 93.6% accuracy	assumption that the performance of a model is transferable
Kalgutkar et al. (2021)	Transfer Learning for the detection of pneumonia from chest X-rays with VGG16-94.07% accuracy	no detailed explanation of the hyperparameters

3 Methodology

CRISP-DM was chosen over KDD for our AutoML library's development because of its industry acceptance and versatility. Figure 1 shows the steps for CRISP-DM, source Wikipedia[2]. CRISP-DM's structured approach, systematic phases, and emphasis on business understanding make it suitable for creating a user-centric AutoML library for transfer

[2] `https://en.wikipedia.org/wiki/File:CRISP-DM_Process_Diagram.png`

learning. Its transparency and documentation requirements ensure the library's development process is well-documented and easily reproducible. CRISP-DM aligns with the complex nature of AutoML library development, making it a natural choice for this project. Lets discuss it in detail below

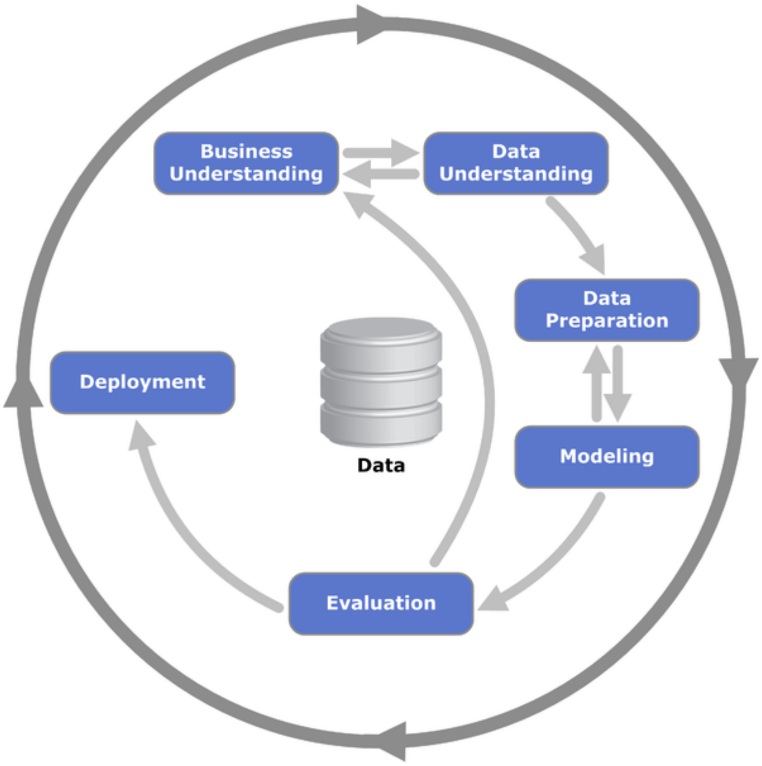

Figure 1: Cross-industry standard process for data mining (CRISP-DM) methodology.

3.1 Business Understanding

In today's data-driven world efficiently using machine learning techniques is crucial. Transfer learning has become important in machine learning for applying knowledge from one task to another saving time and resources. Many businesses, especially those in image and text analysis, natural language processing, and computer vision need transfer learning to stay competitive. This thesis introduces an AutoML library for transfer learning. The library simplifies the use of pre-trained models like VGG19, ResNet-50, Imagenet, DenseNet-121 and EfficientNet-B7 on user datasets. It enables businesses and researchers to deploy these models for classification tasks and obtain accuracy, precision, recall, and F1 score insights. The library simplifies the use of transfer learning methods, helping businesses improve decision-making, products, user experiences, and competitive advantage in AI. Its development can lead to breakthroughs in healthcare, autonomous vehicles, e-commerce, and content recommendation, opening possibilities for innovation and growth.

3.2 Data Understanding

We can use any image classification datasets for the use cases. We selected emotion and pneumonia classification.

3.2.1 Emotion detection dataset

The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centred and occupies about the same amount of space in each image. We have downloaded the CSV files from Kaggle [3] and uploaded in our Google drive.The dataset contains integer codes as labels for emotions, and it has been carefully annotated. This improves the accuracy of machine learning models that classify emotions from facial expressions. The 'pixels' column in the FER dataset is crucial as it holds grayscale pixel information for each image. These pixel values record intensity variations within the images, offering valuable data needed to understand subtle emotional cues. Through analyzing this pixel data, I aim to gain insights into the complex connections between different facial characteristics and the emotions they convey.

Figure 2: FER emotion dataset

3.2.2 Pneumonia Detection dataset

The dataset[4] is organized into 2 files (data and label) and contains image category (Pneumonia/Normal). There are 5,863 X-Ray images and 2 categories (Pneumonia/Normal). Chest X-ray images (anterior-posterior) were selected from retrospective cohorts of pediatric patients of one to five years old from Guangzhou Women and Children's Medical Center, Guangzhou. All chest X-ray imaging was performed as part of patients' routine clinical care. For the analysis of chest x-ray images, all chest radiographs were initially screened for quality control by removing all low quality or unreadable scans. The diagnoses for the images were then graded by two expert physicians before being cleared for training the AI system. In order to account for any grading errors, the evaluation set was also checked by a third expert.

3.3 Modeling

The CRISP-DM model's modeling step involves using a code that establishes various machine learning models. The code tests different pre-trained deep learning architectures

[3] https://www.kaggle.com/datasets/deadskull7/fer2013

[4] https://www.kaggle.com/competitions/rsna-pneumonia-detection-challenge/data

such as VGG19, ResNet-50, Imagenet, DenseNet-121 and EfficientNet-B7 to evaluate how well they perform. The first step involves dividing the dataset into training and validation sets. Then the image data is standardized by rescaling it to values ranging from 0 to 1. Data augmentation techniques like rotation, shifts in width and height, shear, zoom, and horizontal flip are applied. Each deep learning model is constructed by combining a custom output head with a pre-trained architecture from TensorFlow's Keras applications such asVGG19, ResNet-50, Imagenet, DenseNet-121 and EfficientNet-B7. The models undergo training using early stopping and learning rate reduction callbacks while employing the categorical cross-entropy loss function and the Adam optimizer. The training progress, encompassing loss and accuracy is documented. After the completion of training the models are employed to produce predictions on the validation set, and the outcomes are assessed for each model in terms of accuracy, precision, recall, and F1 score. The data table presents the evaluation metrics of each model which are arranged in descending order based on accuracy. These metrics are displayed in a Pandas DataFrame to facilitate comparison between different deep learning models. By considering factors such as accuracy and computational efficiency the best model can be chosen from this table. This step aids in selecting the most appropriate model architecture for the given task, thereby guiding subsequent refinement and optimization in the CRISP-DM process.

3.4 Evaluation

The code provided evaluates the performance of five pre-trained deep learning models (VGG19, ResNet-50, Imagenet, DenseNet-121 and EfficientNet-B7) in the evaluation step of the CRISP-DM model. It starts by dividing the data into training and validation sets and normalizing the image data. Each model is trained using transfer learning with pre-trained architectures and goes through data augmentation, early stopping, and learning rate scheduling during training. Once trained the code calculates important performance measures such as accuracy, precision, recall, and F1 score for each model on the validation set.The results of the evaluation are saved for each model and utilized to create a performance table which is organized as a pandas DataFrame. This table arranges the models based on accuracy, enabling an easy comparison of their performance. The evaluation process assists in choosing the most appropriate model for future stages of the CRISP-DM process like deployment or additional optimization.

3.5 Deployment

The code presented in the deployment phase of the CRISP-DM model showcases the construction and training of several pre-trained deep learning models. The code utilizes a range of pre-existing convolutional neural networks (CNNs), such as VGG19, ResNet-50, Imagenet, DenseNet-121 and EfficientNet-B7. Each individual model is generated, trained and assessed based on its effectiveness in processing the provided image data.Our custom package, py_TransferLearning_Package, can be deployed on GitHub and easily installed using pip. This streamlined process allows users convenient access to the package for various applications. The main focus is on seamlessly transitioning from development to practical implementation while enabling updates and improvements based on performance feedback. The deployment phase in the CRISP-DM process is pivotal, involving saving the model, environment setup, adaptation, application integration, and rigorous testing. Continuous monitoring and maintenance are crucial for sustained high performance. This strategy ensures that the trained models are effectively utilized in real-world applications, allowing for continuous monitoring, adaptability, and optimization whenever required. The process of deployment includes the division of data into training and validation sets, modifying the data, configuring hyperparameters like batch size and epochs, and tracking training progress through callbacks. The code trains each model using the prepared image data and calculates performance measures such as accuracy, precision, recall, and F1 score for each model on the validation data. These measures are then organized in a DataFrame, sorted by accuracy, and returned for comparing and selecting models based on their performance. This step of deployment is essential for implementing the trained models in real-life situations.

4 Design Specification

As shown in Figure 3, our newly developed Auto-ML library starts with the input dataset. It then performs some preprocessing tasks to convert it into a numpy array. Once the dataset has been pre-processed, it is passed to our automl library, which handles data splitting into training and testing sets. In the model training phase, we apply various pre-defined models such as VGG19, ResNet-50, Imagenet, DenseNet-121 and EfficientNet-B7. To ensure effective training and fine-tuning of these pre-defined models, we utilize Callbacks such as EarlyStopping and ReduceLROnPlateau. These Callbacks play a crucial role in overseeing and managing the training process.To enhance the diversity of our training examples we employ ImageDataGenerator techniques. This involves applying data augmentation methods such as rotation, shifting, shearing, zooming, and horizontal flipping to the images. Each pre-defined model is created with specific settings including the choice of loss function, optimizer (Adam with a customized learning rate), and evaluation metric (accuracy). Once all models have been trained and evaluated individually using these metrics a dataframe is generated to organize their performance metrics in descending order based on accuracy. This arrangement simplifies the comparison process so that we can easily identify the top-performing pre-defined model for emotion classification. Details implementation step is mention in Section 5.

5 Implementation

5.1 Input Dataset

For our thesis we have used facial emotion detection and Pneumonia Detection dataset to test our Auto-ML library.we can use any classification dataset on which we have to apply this models and find which one performs best.

5.1.1 Pre-processing of facial emotion detection

The code transforms the string pixel values of each image into numerical values. It arranges these values into a 48x48 grid to represent the image. The values are then converted to a suitable numerical data type for further processing. The code guarantees that every image is in the RGB format rather than grayscale. It accomplishes this by converting each grayscale image into a full-color RGB format. This process assists in preparing the images for utilization with models that require color images. One method

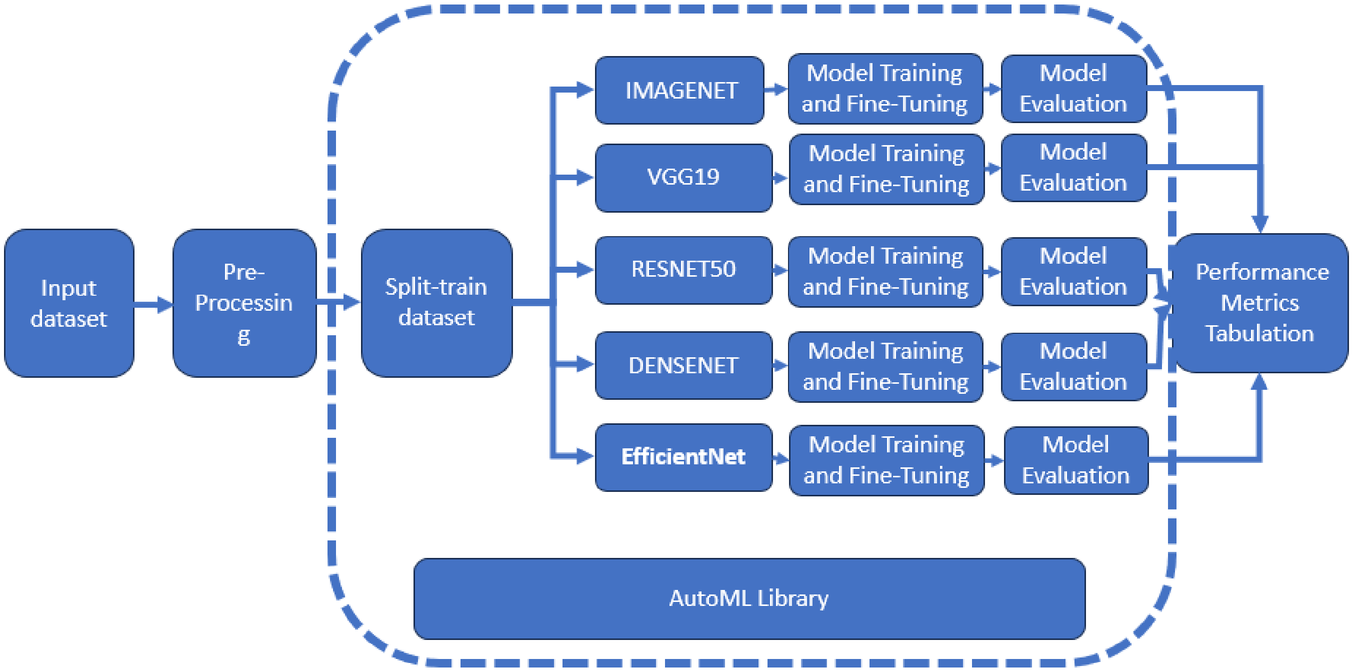

Figure 3: Design flowchart of proposed AutoML library

of encoding emotion labels in image analysis is through label encoding. This involves assigning numeric values to emotions such as “happy”, “sad”, or “angry”. Each emotion is mapped to a distinct number, and these numeric labels are then converted into a one-hot encoded format. In this format, each emotion is represented by a binary vector. Label encoding is frequently employed in training machine learning models to identify emotions in images.The code establishes a label mapping that links the initial emotion class names (such as “happy” or “angry”) to their corresponding numerical labels. This mapping enables comprehension of the model’s output by converting numeric predictions back into the original emotion labels.These preprocessing procedures ready the dataset for utilization for our auto-ml library.

5.1.2 Pre-processing of Pneumonia Detection dataset

The Preprocessing carries out various operations for image processing and data transformation. It begins by creating a blank NumPy array named resized_images with dimensions (number of images in X, 48, 48, 3). Next a loop in which each image is resized to a size of 48x48 pixels using the cv2.resize function from the OpenCV library. The resized images are then saved in the resized_images array.next we transforms the desired labels Y into a NumPy array and computes the count of distinct values within Y denoting the number of unique classes or categories in the dataset. Then we generates a binary vector representation of the labels using the np.eye function which essentially converts the categorical labels into binary vectors where each class is assigned a distinctive bit pattern. The resulting one-hot encoded labels are saved in the labels variable and generates the one-hot encoded labels which offer a binary depiction of the classes for machine learning training. These preprocessing procedures ready the dataset for utilization for our auto-ml library.

5.2 Auto-ML library

We have design a AutoML library in which we have developed two function build model and modeling. The library takes classification dataset and run various pre-trained Keras models (VGG19, ResNet-50, Imagenet, DenseNet-121 and EfficientNet-B7) and gives a result in form of table of accuracy, precision, recall, and F1 score. The Link to our code and artifacts is on Github [5] it is also available in google drive [6]

5.2.1 Built_model function

This functions has two input parameter bottom_model and classes. bottom_model take a preexisting neural network model, typically a pretrained model like VGG19, ResNet-50, Imagenet, DenseNet-121 and EfficientNet-B7 and classes take the number of output classes for the classification task. we use built_model function in modeling function

5.2.2 Modeling function

This function has 4 parameter img_features,img_labels ,batch size and epochs. img_features takes a numpy array of image data as input, img_labels it take a numpy array of target labels,batch size is defaulted 32 as Hyperparameter specifying the batch size for training and epochs is defaulted to 25. The high-level API for the modeling method is:

```
import py_TransferLearning_Package
performance_table = py_TransferLearning_Package.modeling(img_features,
            img_labels, batch_size=32, epochs=10)
```

where batch_size and epochs are optional parameters.

5.2.3 VGG19, ResNet-50, Imagenet, DenseNet-121 and EfficientNet-B7

We utilised pre-trained Keras transfer learning models. Our sample dataset and labels must be used to retrain the models. After retraining we use the model on unseen photos. We created training and validation sets using Python's train_test_split function. We validated 10% of samples which is reported in classification accuracy.

VGG19 is a well-known convolutional neural network structure that is valued for its simplicity and effectiveness. It is composed of 19 layers, including a series of convolutional layers and max-pooling layers. To utilize VGG19, the code loads the pretrained model and replaces its top classification layer with a Global Average Pooling layer and a Dense layer that uses softmax activation for classification purposes. Training involves implementing early stopping and learning rate scheduling. Data augmentation techniques are applied using ImageDataGenerator. Model performance on the validation set is evaluated using accuracy, precision, recall, and F1 score.ResNet (Residual Network) is another deep neural network architecture that addresses the problem of vanishing gradients through skip connections. ResNet-50 refers to a variant of ResNet with 50 layers.Similar to VGG19, the code loads a pretrained ResNet-50 model and applies the same modifications as before. Training procedures remain consistent across both models, including

[5] https://github.com/Rai-Vivek-Engineer/x21142157_AutoML_library

[6] https://drive.google.com/drive/folders/1lfewgnpvPQeJV2ks6WC6Bsrwa5sZcpjp?usp=drive_link

early stopping, learning rate scheduling, data augmentation, and evaluation processes. MobileNet is specifically created for vision applications on mobile devices and embedded systems. Its architecture is lightweight, ensuring a compromise between accuracy and the size of the model.The code loads the MobileNet model that has already been trained, makes modifications to it, and trains it using the same methods as the previous models. EfficientNet is a group of models that aim to improve model efficiency by optimizing the architecture. EfficientNetB7 is one of the largest versions within this family. The code loads the pre-trained EfficientNetB7 model, makes changes to it, and trains it using similar methods as the other models. DenseNet (Densely Connected Convolutional Networks) promotes the reuse of features by establishing connections between each layer and all subsequent layers. DenseNet-121, a variant consisting of 121 layers, is utilized in this scenario. The provided code loads the pre-trained DenseNet-121 model and makes necessary modifications before training it using similar procedures as previous models. For every model, the code implements data preprocessing, augmentation, early stopping and learning rate scheduling to train the model on the training data. Subsequently, it assesses the performance of each model on the validation set by considering metrics such as accuracy, precision, recall, and F1 score.ImageDataGenerator from TensorFlow's Keras API is used. Shear, zoom, horizontal flip, rotation, width and height shift ranges are set in train_datagen. It creates augmented batches on-the-fly by randomly transforming input photos during training. These transformations vary the training set by rotating, shifting, shearing, zooming, and horizontal flipping. Data augmentation improves the model's generalisation by using additional training instances.

5.3 Performance metrics tabulation

The code for tabulating performance metrics creates a summary table that displays the evaluation metrics of each of the five pre-trained deep learning models. This table is useful for comparing and choosing the best-performing model. The code calculates various important performance metrics, such as accuracy, precision, recall and F1 score for each model on the validation dataset. Then it arranges these metrics into a Pandas DataFrame where each row corresponds to a different pre-trained model (VGG19, ResNet-50, Imagenet, DenseNet-121 and EfficientNet-B7). The DataFrame includes columns for the model names as well as accuracy, precision, recall, and F1 score. The models are sorted in descending order of accuracy to easily identify the most accurate one. This summarized table acts as a valuable tool for selecting models and enables data scientists and researchers to make informed decisions about which deep learning architecture performs optimally for a specific task or dataset.

6 Evaluation

The evaluation metrics, such as classification accuracy, precision, recall, and F1 Score, for different pre-trained deep learning models applied to a specific task. Metrics are reported from the validation set. The models listed in both tables are VGGNet19, DenseNet-121, EfficientNet-B7, ResNet50, and MobileNetV2. These metrics reflect the models' performance in predicting or classifying data, where higher values are indicative of better performance.

Model	Accuracy	Precision	Recall	F1 Score
VGGNet19	0.685985	0.689581	0.685985	0.684925
DenseNet-121	0.662023	0.665288	0.662023	0.661796
ResNet50	0.639454	0.651272	0.639454	0.642530
EfficientNet-B7	0.603511	0.634531	0.603511	0.611310
MobileNetV2	0.595709	0.628730	0.595709	0.596789

Figure 4: Sample output from the emotion classification use case

Model	Accuracy	Precision	Recall	F1 Score
VGGNet19	0.800000	0.796167	0.800000	0.797786
MobileNetV2	0.781818	0.770563	0.781818	0.752066
DenseNet-121	0.763636	0.821612	0.763636	0.689782
ResNet50	0.727273	0.528926	0.727273	0.612440
EfficientNet-B7	0.727273	0.528926	0.727273	0.612440

Figure 5: Sample output from the Pneumonia classification use case

6.1 Discussion

In Figure 4 we can see that VGGNet19 displayed the highest Accuracy at around 68.59%, followed closely by DenseNet-121 with roughly 66.05%. EfficientNet-B7, ResNet50, and MobileNetV2 achieved slightly lower accuracies of 59.57%. In terms of Precision VGGNet19 performed the best reaching approximately 68.95%, whereas DenseNet-121 demonstrated a precision of about 66.52%.

Figure 5 shows that VGGNet19 retained its high performance, showing an increased accuracy of 80%, and maintaining precision, recall, and F1 Score around 79.72% to 80%. MobileNetV2 followed closely with an accuracy of about 78.18%, showing a relatively balanced Precision and Recall. DenseNet-121 exhibited an accuracy of approximately 76.36%, with a higher precision of 82.16%, yet a relatively lower F1 Score. ResNet50 and EfficientNet-B7 had the same accuracy and almost identical Precision, Recall, and F1 Score values, standing at 72.73%.

Interestingly, in both use cases VGG19 outperformed other more complex models. In emotion recognition the different was much better than the others, but in the pneumonia classification the results were very close to the others.

We have develop an automl library that can take a classification dataset and run various Keras pre-trained model like VGGNet19, DenseNet-121, EfficientNet-B7, ResNet50, and MobileNetV2 and gives result inform of table with accuracy, precision, recall, and F1 score which answer our Research question 1 (RQ1) that we were able to automate it to a good extend that even a user who know basic coding can use our library and run various transfer learning pre-trained models. Section 5.2 of our paper helps us to answer our research question 2 (RQ2) and gives a detail idea how and automl library was developed. section 2 also answer our Research objectives 1 and Research objectives 4. Section 5.2.3 answer the Research objective 2 where we talk about the different keras pre-trained models and how we used in our own library it also answer to our Research obejective 3.

7 Conclusion and Future Work

The importance of automation and accessibility in the current era of machine learning is emphasized in this study which introduces a new AutoML library that makes it easier to use pre-trained transfer learning approaches on different datasets. The work addresses a very important research gap in the machine learning community. We successfully developed a novel AutoML library to automate transfer learning workflows. To test the validity of this new library we used two use cases for image classification. We have used facial emotion and pneumonia classification datasets. These two challenging evaluations have shown that the AutoML library can greatly enhance the user experience in the usage of machine learning models. Our aim is to simplify the use of transfer learning approaches on different datasets. Even a user with basic coding knowledge can use this library. In conclusion, this study contributes to fill in a research gap that found an opportunity for AutoML to make transfer learning more accessible to all. It can also serve as inspiration for future advancements in automated model creation.

It is advised that in the future more testing should be done on this library using a wider range of datasets to evaluate how versatile and robust it is. Up until now the development and testing stages have mainly concentrated on classification datasets however there is potential for it to also be used with regression datasets. Moreover some default parameters in the code are hidden from the user and further extensions of the AutoML library should focus on passing parameters to override these default settings. As part of future efforts enhancing the code to allow greater configurability and adaptability of parameters would be a valuable enhancement.

8 Acknowledgements

I want to acknowledge and thanks Dr. Giovani Estrada for helping me out so much with my thesis research. Dr. Giovani Estrada has given me lot of really helpful advice and feedback that has made a big difference in my work on this paper and code. I've been inspired by his guidance and expertise and I feel really lucky to have had the chance to work with him. I also want to thank my family and all the professors for their invaluable guidance and support.

References

Doke, A. and Gaikwad, M. (2021). Survey on automated machine learning (automl) and meta learning, *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1–5.

Ghosh, A., Rab, R. and Rahman, A. (2022). Transfer learning based facial emotion detection for animated characters, *2022 25th International Conference on Computer and Information Technology (ICCIT)*, pp. 876–881.

Giri, S., Singh, G., Kumar, B., Singh, M., Vashisht, D., Sharma, S. and Jain, P. (2022). Emotion detection with facial feature recognition using cnn & opencv, *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pp. 230–232.

Han, B., Yoo, C.-H., Kim, H.-W., Yoo, J.-H. and Jang, J. (2023). Deep emotion change detection via facial expression analysis, *Neurocomputing* **549**: 126439.
URL: *https://www.sciencedirect.com/science/article/pii/S0925231223005623*

He, K., Zhang, X., Ren, S. and Sun, J. (2015). Deep residual learning for image recognition.

He, X., Zhao, K. and Chu, X. (2021). Automl: A survey of the state-of-the-art, *Knowledge-Based Systems* **212**: 106622.

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

Huang, G., Liu, Z., van der Maaten, L. and Weinberger, K. Q. (2018). Densely connected convolutional networks.

Jaiswal, A., Krishnama Raju, A. and Deb, S. (2020). Facial emotion detection using deep learning, *2020 International Conference for Emerging Technology (INCET)*, pp. 1–5.

Kalgutkar, S., Jain, V., Nair, G., Venkatesh, K., Parab, K., Deshpande, A. and Ambawade, D. (2021). Pneumonia detection from chest x-ray using transfer learning, *2021 6th International Conference for Convergence in Technology (I2CT)*, pp. 1–6.

Khajontantichaikun, T., Jaiyen, S., Yamsaengsung, S., Mongkolnam, P. and Ninrutsirikun, U. (2022). Emotion detection of thai elderly facial expressions using hybrid object detection, *2022 26th International Computer Science and Engineering Conference (ICSEC)*, pp. 219–223.

Kotlar, M., Punt, M., Radivojević, Z., Cvetanović, M. and Milutinović, V. (2021). Novel meta-features for automated machine learning model selection in anomaly detection, *IEEE Access* **9**: 89675–89687.

Salehin, I., Islam, M., Saha, P., Noman, S., Tuni, A., Hasan, M. and Baten, M. (2023). Automl: A systematic review on automated machine learning with neural architecture search, *Journal of Information and Intelligence* .
URL: *https://www.sciencedirect.com/science/article/pii/S2949715923000604*

Shao, L., Zhu, F. and Li, X. (2015). Transfer learning for visual categorization: A survey, *IEEE Transactions on Neural Networks and Learning Systems* **26**(5): 1019–1034.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition.

Tan, M. and Le, Q. V. (2020). Efficientnet: Rethinking model scaling for convolutional neural networks.

Xue, C., Yan, J., Yan, R., Chu, S. M., Hu, Y. and Lin, Y. (2019). Transferable automl by model sharing over grouped datasets, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8994–9003.

Yuvaraj, S., Franklin, J. V., Prakash, V. S., Anandaraj, A. and Subha, R. (2023). A survey on human face emotion recognition using machine learning models, *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Vol. 1, pp. 1993–1996.

Zhang, Z., Lin, P., Ma, S. and Xu, T. (2022). An improved yolov5s algorithm for emotion detection, *2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, pp. 1002–1006.