BRAIN-INSPIRED LEARNING FOR FACE ANALYSIS IN ARTIFICIAL NEURAL NETWORKS: A MULTITASK AND CONTINUAL LEARNING FRAMEWORK

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

SEFA BURAK OKCU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF COGNITIVE SCIENCE

JANUARY 2023

Brain-Inspired Learning for Face Analysis in Artificial Neural Networks: A Multitask and Continual Learning Framework

submitted by **SEFA BURAK OKCU** in partial fulfillment of the requirements for the degree of **Master of Science in Cognitive Science Department, Middle East Technical University** by,

Prof. Dr. Banu Günel Kılıç
Dean, **Graduate School of Informatics**

Dr. Ceyhan Temürcü
Head of Department, **Cognitive Science**

Prof. Dr. A. Aydın Alatan
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Assist. Prof. Dr. Umut Özge
Co-supervisor, **Cognitive Science Dept., METU**

Examining Committee Members:

Assist. Prof. Dr. Murat Perit Çakır
Cognitive Science Dept., METU

Prof. Dr. A. Aydın Alatan
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. Erkut Erdem
Computer Engineering Dept., Hacettepe University

Date: 26.01.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Sefa Burak Okcu

Signature :

ABSTRACT

BRAIN-INSPIRED LEARNING FOR FACE ANALYSIS IN ARTIFICIAL NEURAL NETWORKS: A MULTITASK AND CONTINUAL LEARNING FRAMEWORK

Okcu, Sefa Burak
M.S., Department of Cognitive Science
Supervisor: Prof. Dr. A. Aydın Alatan
Co-Supervisor: Assist. Prof. Dr. Umut Özge

January 2023, 66 pages

The phenomenon known as catastrophic forgetting is common in connectionist models while learning from a sequence of data from different distributions. On the other hand, the human brain has the ability to learn from a sequence of experiences continually while retaining old information. Recent studies utilize different brain-inspired methods such as regularization, parameter isolation, and replay to alleviate this problem in artificial systems. Following the previous studies, we investigated different continual learning methods on face analysis tasks involving age estimation, binary gender recognition, emotion recognition, and face recognition. Neurological findings implicate that there are different specialized functional and neural areas in the brain for the perception of faces. Similarly, we analyzed faces in two stages, very common in artificial neural networks: face detection and face attributes analysis. Firstly, experiments for learning face detection and facial landmark detection were conducted by studying multitask learning. Secondly, some continual learning methods inspired by biological systems were leveraged to overcome catastrophic interference in artificial models. In the first experiments, our proposed model was able to learn both face and facial landmark detection efficiently, along with a performance boost. In later experiments, we observed that the utilized continual learning methods performed better on task incremental scenarios than class incremental scenarios. Nevertheless, a combination of two different continual learning methods resulted in remarkable performance improvement in class incremental scenarios. As a result, the combination of different alternative neuroscience-inspired methods is required for mitigating forgetting and approaching multitask performance.

Keywords: continual learning, multitask learning, catastrophic forgetting, face detection, face analysis

ÖZ

YAPAY SİNİR AĞLARINDA YÜZ ANALİZİ İÇİN BEYİNDEN İLHAM ALAN ÖĞRENME: ÇOK GÖREVLİ VE SÜREKLİ ÖĞRENME SİSTEMİ

Okcu, Sefa Burak
Yüksek Lisans, Bilişsel Bilimler Bölümü
Tez Yöneticisi: Prof. Dr. A. Aydın Alatan
Ortak Tez Yöneticisi: Dr. Öğr. Üyesi. Umut Özge

Ocak 2023, 66 sayfa

Bağlantıcı modellerde farklı dağılımlardan gelen veri dizisi üzerinden öğrenme sırasında katastrofik unutma olayı yaygındır. Diğer yandan, insan beyni eski bilgileri saklarken sürekli olarak deneyimler dizisinden öğrenme yeteneğine sahiptir. Son çalışmalar, yapay sistemlerde bu sorunu azaltmak için düzenleme, parametre ayrımı ve yeniden oynatma gibi beyinden ilham alan yöntemleri kullanmaktadır. Önceki çalışmaları takiben yaş tahmini, ikili cinsiyet tanıma, duygu tanıma ve yüz tanıma içeren yüz analizi görevlerinde farklı sürekli öğrenme yöntemlerini inceledik. Nörolojik bulgular, beyinde yüz algılama için uzmanlaşmış işlevsel ve sinirsel alanların bulunduğunu işaret etmektedir. Benzer şekilde, biz bu çalışmada yüz analizini yapay sinir ağlarında da sıklıkla görüldüğü gibi iki aşamada inceledik: yüz tespiti ve yüz özellik analizi. Öncelikle, çoklu görev öğrenimi çalışarak yüz tespiti ve yüz işaret noktaları tespiti üzerine deneyler gerçekleştirdik. İkinci olarak, biyolojik sistemlerden ilham alan bazı sürekli öğrenme yöntemlerini kullanarak yapay modellerdeki katastrofik girişimi aşmayı hedefledik. İlk deneylerimizde, önerdiğimiz model performans artışının yanı sıra yüz ve yüz işaret noktaları tespitini verimli bir şekilde öğrenebildi. Sonraki deneylerimizde, kullanılan sürekli öğrenme yöntemlerinin görev artımlı senaryoda sınıf artımlı senaryoya göre daha iyi performans gösterdiğini gözlemledik. Bununla birlikte, iki farklı sürekli öğrenme yönteminin bir kombinasyonu, sınıf artımlı senaryolarda dikkate değer bir performans artışı sağladı. Sonuç olarak, unutmanın azaltılması ve çoklu görev performansına ulaşılması için farklı alternatif nörobilimden ilham alınan yöntemlerin birleştirilmesi gerekmektedir.

Anahtar Kelimeler: sürekli öğrenme, çok görevli öğrenme, katastrofik unutma, yüz tespiti, yüz analizi

To my precious family

ACKNOWLEDGMENTS

I am deeply grateful to my supervisor, Prof. Dr. A. Aydın Alatan, for his invaluable guidance and support throughout the course of my research. His expertise and encouragement were essential in helping me to develop my ideas and bring my work to fruition. The meetings with him were always productive, thought-provoking, and exciting, and I have learned so much from him. I would also like to express my appreciation to my co-advisor, Asst. Prof. Umut Özge for their valuable guidance.

I would also like to extend my heartfelt thanks to my leader and co-supervisor, Dr. Cevahir Çığla, for providing me with the opportunity to conduct this research as part of my professional development. Their guidance and support have been priceless. He has been a valuable mentor and an idol for me to look up to and learn from.

I cannot express enough gratitude to my family for their unwavering support and encouragement. Without their love and encouragement, this journey would not have been possible. I am also grateful to my friends Yağız Önen and Fiona Polloshka for their constant encouragement and companionship. Their presence has made all the difference in my journey.

I would also like to thank all of the professors, staff, and fellow students who have contributed to my learning and growth during my time in the Cognitive Science program. Their insights, knowledge, and support have been invaluable to me.

Finally, I would like to extend my sincere gratitude to Aselsan Inc. for providing me with the necessary hardware support and facilities during my thesis work. Their contribution was invaluable and greatly facilitated my research.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACC	Average Accuracy
AFLW	Annotated Facial Landmarks in the Wild
ANNs	Artificial Neural Networks
BIR	Brain Inspired Replay
BWT	Backward Transfer
CelebA	CelebFaces Attributes Dataset
CL	Continual Learning
CLIFER	Continual Learning Framework with Imagination for Facial Expression Recognition
CLS	Complementary Learning Systems
CNNs	Convolutional Neural Networks
CPG	Compacting, Picking, and Growing
CSPNet	Cross Stage Partial Network
DCNNs	Deep Convolutional Neural Networks
DEN	Dynamically Expandable Network
DGR	Deep Generative Replay
EBLL	Encoder Based Lifelong Learning
EWC	Elastic Weight Consolidation
FD	Face Detection
FL	Facial Landmark Detection
GR	Generative Replay
HAT	Hard Attention to the Task

iCaRL	Incremental Classifier and Representation Learning
LSTM	Long Short Term Memory
LtG	Learn to Grow
LTP	Long-term Potentiation
LwF	Learning without Forgetting
MAS	Memory Aware Synapses
ML	Machine Learning
MMoE	Multi-gate Mixture-of-Experts
MTCNN	Multitask Cascaded Convolutional Neural Networks
MTL	Multitask Learning
NLP	Natural Language Processing
NMS	Non Maximum Suppression
NRMSE	Normalized Root Mean Square Error
PAE	Packing and Expanding
PaLM	Pathways Language Model
PCA	Principal Component Analysis
RT-1	Robotics Transformer 1
SARSA	State-Action-Reward-State-Action
SGD	Stochastic Gradient Descent
SI	Synaptic Intelligence
SPP	Spatial Pyramid Pooling
VAE	Variational AutoEncoder
YOLO	You Only Look Once

CHAPTER 1

INTRODUCTION

The human brain has the ability to learn continually, whereas connectionist networks are not capable of learning new tasks continually without forgetting old tasks, which is called catastrophic forgetting [7] [8]. In addition, humans can acquire new knowledge when different but related information is provided together, as well as when learning separately.

In this thesis, the performance of multitask learning and continual learning techniques are evaluated on Deep Convolutional Neural Networks (DCNNs) for various face analysis tasks, with the goal of drawing inspiration from the human brain to improve their effectiveness. The focus is on examining the potential of these techniques to enable DCNNs to learn from multiple tasks simultaneously and adapt to new tasks without forgetting previously learned knowledge.

1.1 Problem Definition

Current state-of-the-art DCNNs outperformed humans in some specific tasks such as object recognition [9], translation [10], and game playing [11] [12] while humans are generally good at learning multiple tasks together. Artificial Neural Networks (ANNs) can also be optimized for multiple related tasks jointly for better efficiency and generalization capability when labels are provided for multiple tasks together on the same data. However, data arrives sequentially in nature, and it is not easy to find a labeled data set for multiple tasks. In addition, joint optimization of ANNs on multitasks sometimes leads to degradation of the performance of each task due to interference between the tasks. Thus, multitask learning requires tasks to be related to each other. In addition to task relatedness, there are different problems; such as how to combine losses from tasks with different weights and how much information should be shared between tasks, which require extra work in multitask learning literature.

Continual Learning (CL) is another method that mammalian brains exhibit for learning and can be an alternative for overcoming the problems in Multitask Learning (MTL) described above for training ANNs. In contrast to biological neural networks, ANNs cannot incrementally learn new tasks and utilize information gained from one task for faster and better learning of new tasks straightforwardly without catastrophically forgetting. It occurs when an individual or system struggles to balance the conflicting demands of plasticity and stability, which is called the stability-plasticity dilemma [13]. This conflict arises because the brain must maintain a certain level of stability in order to support the retention of learned information while also remaining flexible enough to facilitate the acquisition of new information and adapt to changing environments. Although different methods are proposed for overcoming the stability-plasticity dilemma in CL literature, most of the experiments are conducted on

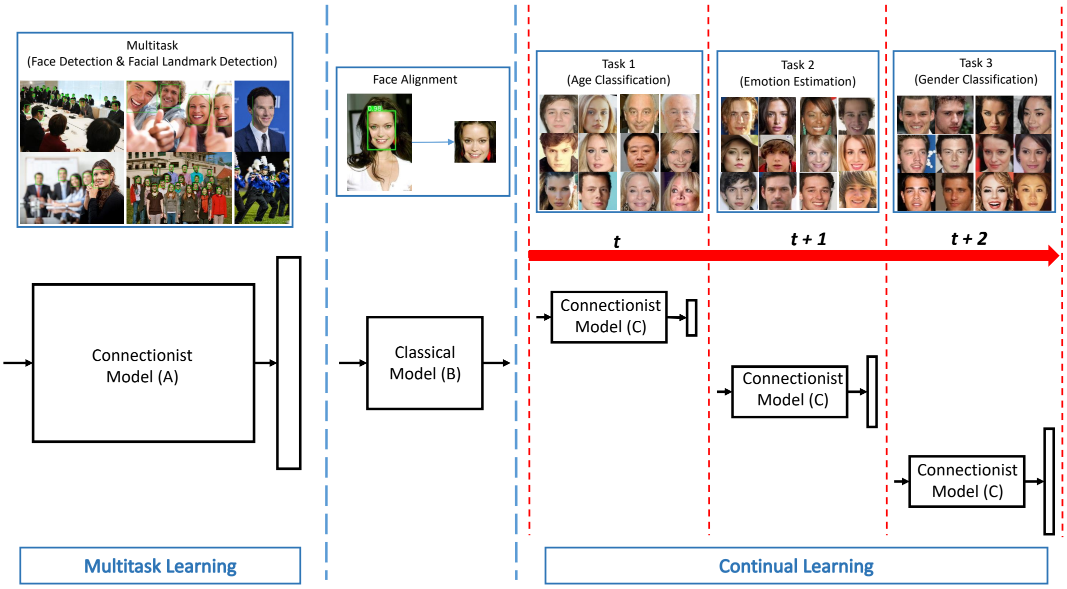


Figure 1: An overview of our proposed Multitask and Continual Learning framework for face analysis in Artificial Neural Networks.

classification tasks, and relatively easy and solved datasets such as MNIST [14] and CIFAR100 [15]. In addition, the resolution and the number of images are relatively small compared to the samples used in other Machine Learning (ML) paradigms.

In order to examine MTL and CL, we utilized face analysis in our experiments. Evidence from behavioral, neuropsychological, and neurophysiological studies suggests the face-specificity hypothesis, which proposes specialized cognitive and neural areas for preferably processing faces [16] [17] [18] [19]. According to findings of Kanwisher [20] from experiments using functional magnetic resonance imaging, there is a cortical region specialized for the perception of faces [21], which is called the fusiform face area. Furthermore, Bruce and Young [22] proposed a cognitive model for face processing, which divides face processing into different functional levels. In their system, Face Detection (FD) was the essential initial step for processing faces at later levels. Moreover, Farah et al.[23] discovered that prosopagnosics, who lose the ability of face recognition, are better than normal people in the recognition of inverted faces. In addition to evidence for the separation of face perception in human brains, Tsao and Livingstone indicated that researchers analyze faces in three stages, which are detection, measurement, and categorization using computer vision in their review paper [24]. As a result, we also employed face-related tasks such as face detection, facial landmark extraction, age estimation, gender recognition, emotion estimation, and face recognition in our MTL and CL experiments. We also followed a similar process that divides face analysis into two specialized neural and functional modules: one for face detection and Facial Landmark detection (FL), and another one for age, gender, and emotion recognition or face recognition (Figure 1).

1.2 Scope of the Thesis

Face analysis requires the detection and analysis of each face in the scene. In the human brain, different specialized cognitive levels and functions possibly exist for detecting and recognizing faces [19]. The fusiform face area, which is specialized in the perception of faces, is also activated when other facial tasks such as age, gender, and emotion recognition are performed in the human brain [25] [21]. In this thesis, like human brains, we split the face analysis problem in ANNs into two stages, detection of faces and analysis of each detected face, and propose two connectionist models for each stage. In the first part, we examine the effects of MTL on face detection and facial landmark detection tasks by utilizing the Widerface [1] dataset, which has samples from real-world scenarios and labels for both tasks. We conduct experiments in order to understand the effects of optimization of FD and FL tasks jointly and separately by utilizing our proposed DCNNs architecture. Furthermore, we extend the model's capability by allowing it to learn the second task after the first task is acquired. In the second part, we employ different CL methods, including a brain-inspired replay method [4] for analyzing their performance on separate face analysis tasks, which are emotion, gender and age recognition, according to a task incremental learning scenario. The experiments are performed on CelebFaces Attributes Dataset (CelebA) [3] dataset after our MTL model is applied to each image to obtain aligned faces. Additionally, we compare the same CL methods for face recognition by splitting the total number of identities into several episodes/tasks. Experiments are designed according to both task and class incremental learning scenarios on VGGFace2 [6] dataset.

1.3 Outline of the Thesis

In this thesis, the related background is presented in Chapter 2. Initially, different machine learning paradigms are explained briefly. After a description of machine learning paradigms, an overview of the literature on multitask learning and continual learning is presented. We present past general works briefly and the methods related to the scope of this thesis are thoroughly explained. We begin by reviewing comprehensive literature on MTL, then delve into more specific studies focused on face analysis. Afterward, we introduce different CL methods under three main categories. In Chapter 3, experimental setups for evaluating the effects of MTL are described in detail. Later, the results of single and multitask models are presented on face detection and facial landmark detection tasks. In Chapter 4, different continual learning methods are evaluated on realistic face data sets. The results are compared on face analysis tasks considering class incremental and task incremental learning scenarios. Finally, the summary of the thesis and its concluding remarks are presented in Chapter 5.

CHAPTER 2

RELATED BACKGROUND

In this chapter, different machine learning paradigms and the literature related to the scope of this thesis are presented. In the first section, some learning paradigms utilized in ML literature are given briefly. After a brief introduction to different learning paradigms in ML, a general overview of methods in multitask learning and continual learning literature are presented concisely in the literature survey section, whereas methods and approaches employed in our experiments or related to our works are examined comprehensively.

2.1 Learning Paradigms in Machine Learning

This section covers a range of machine learning paradigms, including both traditional and modern approaches, which are:

- Supervised learning,
- Unsupervised learning,
- Self-supervised learning,
- Reinforcement learning,
- Transfer learning,
- Domain adaptation,
- Knowledge distillation,
- Curriculum learning,
- Meta-learning,
- Multitask learning,
- Continual learning.

The field of machine learning draws inspiration from the human brain's method of processing information and aims to mimic and improve cognitive abilities. Certain machine learning approaches emulate the human brain's ability to learn and adjust to new information. The versatility of these approaches is vital as it allows them to tackle a wide variety of issues and datasets within machine learning. Familiarizing oneself with the different approaches is crucial for determining the most appropriate algorithm or method for a particular machine learning task. Therefore, the following subsections present those approaches briefly.

2.1.1 Supervised Learning

Supervised learning is a machine learning paradigm in which a model is trained on a labeled dataset, where the correct output is provided for each input sample [26]. The model learns to predict the output for new, unseen input samples by mapping the input to the correct output through a training process [27]. Classic examples of supervised learning include a model that is trained to recognize handwritten digits based on images and their corresponding labels (i.e., the digits 0 through 9) [28], or a model trained to classify images of dogs and cats. Other common applications of supervised learning include predicting the price of a house based on a labeled dataset of house prices and their corresponding features (e.g., number of bedrooms, square footage, location) [29]. Some well-known algorithms for supervised learning involve support vector machines [30] and neural networks [31].

2.1.2 Unsupervised Learning

Unlike supervised learning, which involves training a model on a labeled dataset, in unsupervised learning [26], a model is trained on an unlabeled dataset and must discover patterns and relationships within the data without guidance. This type of learning allows the model to explore the underlying structure of the data and extract meaningful insights [27]. One example of unsupervised learning is clustering, in which a model groups similar data points together based on their characteristics [32]. Another example is dimensionality reduction, in which a model reduces the number of features in a dataset while maintaining as much of the original information as possible [33]. Some well-known algorithms for unsupervised learning include k-means clustering [34] and principal component analysis [35]. For instance, a model could be trained to group similar customer reviews together based on their content using clustering [36], or to reduce the dimensionality of a dataset of stock prices using Principal Component Analysis (PCA) [37].

2.1.3 Self-supervised Learning

High-quality data should be labeled for training in supervised learning, whereas unsupervised learning enables a model to learn the underlying structure of the data or common representations between examples. On the other hand, self-supervised learning utilizes some part of the data in order to learn the other unseen part of the data. Self-supervised learning is also called pretext learning. It turns unsupervised learning into supervised learning by extracting or generating labels automatically. Self-supervised learning is usually preferred as an initial step for learning the representation of data, and then the trained model is trained further for a downstream task. For instance, obtaining a colorful image from a grayscale image [38], learning representation using geometric transformation [39], predicting the hidden word from other words in a sentence, and generating images using generative adversarial networks are some examples in that self-supervised learning is used.

2.1.4 Reinforcement Learning

Reinforcement learning is a type of machine learning where an agent learns to make decisions by interacting with its environment to get the best outcome [40]. The agent gets feedback in the form of rewards or penalties based on its actions and improves its decision-making by experimenting and trying different options [41]. It has been used to solve a variety of problems such as control systems, games and natural language processing [33]. Some examples of popular algorithms for reinforcement learning include Q-learning [42] and State-Action-Reward-State-Action (SARSA) [43]. One application of this type of learning could be training an agent to play a video game by making choices that increase its score [44], or to control a self-driving car by taking actions that enhance its safety and performance [45]. Additionally, reinforcement learning has been used to optimize industrial processes, train robots, and improve decision-making in financial markets.

2.1.5 Transfer Learning

Transfer learning is a very common technique in which a model trained on one task is fine-tuned for use on a different but related task [46]. This approach can be especially useful when there is a limited amount of data available for a specific task. Since it allows the model to leverage its prior knowledge from the original task to improve its performance on the new task [47]. In addition, transferring old knowledge also reduces the time required for the model to learn new tasks. Transfer learning has been involved in many different fields, including natural language processing and computer vision. For example, a model trained to classify hundreds of different objects could also be used for training and classifying images of flowers. One very common instance of a transfer learning algorithm is the convolutional neural network architecture [28], which is commonly used for image classification tasks and is initially trained on Imagenet [48] dataset and fine-tuned for a new image classification task using a small amount of labeled data.

2.1.6 Domain Adaptation

Domain adaptation refers to the process of adapting a model trained on one domain (source, e.g., a specific dataset or task) for use in a different but related domain (target). This technique is often employed when there is a significant difference between the training and test distributions, as it allows the model to generalize better to the new domain. Although both domain adaptation and transfer learning are utilized for the adaptation of models, transfer learning, which refers to the process of adapting a pre-trained model for a new task by fine-tuning the model on a new dataset, is often used when there is a limited amount of labeled data available for the new task. It allows the model to leverage the knowledge learned from the previous task to improve performance on the new task. On the other hand, domain adaptation is useful for the same tasks from different distributions. For instance, a model trained to perform sentiment analysis on reviews of one product could be adapted for sentiment analysis on reviews of another product [49]. Transfer learning is a related but distinct concept from domain adaptation, as it involves adapting a model to a new task rather than a new domain.

2.1.7 Knowledge Distillation

Another machine learning paradigm is knowledge distillation which is used to enhance the performance of smaller models. Although the computation power of the devices for running large models increases, the deployed models are usually preferred as being small and efficient. Therefore, knowledge distillation transfers the existing knowledge from a large model or ensemble of the models, teacher models, to a smaller, compressed model or a student model. This is achieved by the training of the student model on the same data set that the larger model was trained on, and the predictions of the large model are utilized as soft labels for the student model. One generalized method for knowledge distillation is the distillation method introduced by Hinton et al. in 2015 [50]. In the distillation method, the student model aims to learn the teacher model's output probabilities by minimizing the distillation loss. Additionally, the student model also employs its own classification loss in the classification problem, and both losses are combined with different weights.

2.1.8 Curriculum Learning

Curriculum learning [51] is another machine learning paradigm that is utilized for training a model using data in a meaningful order instead of a randomly shuffled order. Humans are able to learn more effectively and efficiently if tasks are supplied according to their difficulty. When the given examples become harder gradually, humans can accumulate knowledge better. Similarly, curriculum learning shortens the time that a training process converges during the training of a model. Furthermore, it also improves the generalization performance of the trained model. One example task is object detection where curriculum learning is useful in which a model is trained with samples that are visible, large, and clear for detection, and later difficulty of training data is gradually increased by the addition of complex, occluded, low-resolution, and small examples.

2.1.9 Meta-Learning

Meta-learning, which is also known as learning to learn, is a machine learning paradigm that facilitates the learning or adapting of the model to new tasks or environments effectively. In order to achieve it, the meta-learning algorithm takes metadata as input and optimizes the learning of the main task solver model by utilizing the performance of the task solver model. It enables the model to adapt to new tasks and environments without using large training data sets. In addition, meta-learning reduces the number of experiments for obtaining good performances. The aim of meta-learning is to adapt and acquire new tasks quickly, similar to people who have the ability to learn from a few samples in addition to faster adaptation to new environments. Moreover, it has several advantages, such as faster training, higher performance, etc. One method of meta-learning is Model-Agnostic Meta-Learning (MAML) which is a task-agnostic algorithm that facilitates the learning of the model faster by small gradient update on a few samples [52].

2.1.10 Multitask Learning

Multitask learning is a type of machine learning in which a single model is trained to perform multiple tasks simultaneously [53]. This approach is inspired by the ability of the human brain to perform multiple tasks simultaneously and to learn from multiple sources of information. In a similar way, multitask learning in machine learning involves training a single model to perform multiple tasks simultaneously and to learn from the shared information among the multiple tasks and environments more efficiently.

There is some evidence that multitask learning can be more effective than training a separate model for each task, particularly when the tasks are related and share some common underlying information [53]. This is thought to be because multitask learning allows the model to learn from the shared information among the tasks effectively and to make use of the commonalities among the tasks to improve its overall performance, which is also particularly useful when there is a limited amount of labeled data available for each task, as the model can learn from the larger combined dataset. For example, a single model could be trained to perform both image classification and object detection using a multitask loss function that combines the losses for both tasks. Alternatively, a single model could be trained to perform both sentiment analysis and topic classification on text. Recently, several different multitask models are developed, such as Pathways Language Model (PaLM) [54] for multiple language tasks i.e. multilingual tasks and generation, Gato [55] for multiple broad tasks (playing Atari, captioning images and chatting, etc.), and Robotics Transformer 1 (RT-1) [56] for over 700 different tasks.

2.1.11 Continual Learning

The ability of a machine learning model to adapt and learn new tasks and environments over time, rather than being trained on a fixed set and then deployed, is known as continual learning (aka. Lifelong learning or incremental learning). This is essential for real-world scenarios, as the model's tasks and surroundings may change. Algorithms for continual learning are created to learn new tasks without forgetting previous ones, a problem known as catastrophic forgetting [57] [58]. One benefit of continual learning is that the model can improve its performance on new tasks by building on the knowledge and skills acquired from previous experiences instead of retraining the model from scratch, which can be both time-consuming and demanding on resources. Continual learning is ideal for dynamic situations where data is constantly changing, such as streaming data.

The human brain is capable of continually learning and adapting to new information, as evidenced by our ability to learn new languages, skills, and even physical abilities [59]. This ability is closely connected to the concept of neural plasticity, which refers to the brain's ability to change and adapt in response to new experiences and learning. Similar to multitask learning, continual learning enables the model to realize multiple tasks. However, continual learning differs from multitask learning, where a single model is trained to perform multiple tasks simultaneously with the goal of improving performance on all tasks by sharing information among them. On the other hand, the focus of continual learning is on training a model to learn and adapt to new tasks and environments over time without forgetting previous knowledge. An example of this is how a student can learn multiple subjects in school but still retain and apply information learned in previous grades.

2.2 Literature Survey of CL and MTL

In this section, a general overview of MTL and CL literature is given. In the first section, different multitask learning techniques are presented briefly. MTL methods that are not problem specific

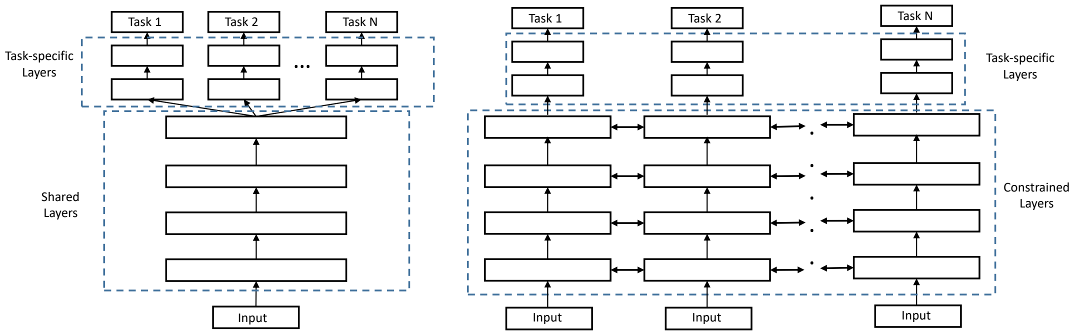


Figure 2: Hard and soft parameter sharing in CNNs.

are introduced. In the second section, CL methods for mitigating catastrophic forgetting are presented. Although methods are divided into three categories which are regularization-based methods, parameter isolation-based methods, and replay-based methods; they have no clear boundaries and can overlap to some extent. While a general overview of MTL and CL strategies is presented in the previous sections, the techniques and strategies applied specifically to face analysis, including face detection, facial landmarks detection, age estimation, emotion estimation, gender recognition, and face recognition tasks, are included in detail in the third section.

2.2.1 Multitask Learning

In this section, MTL techniques for architectural implementation are given briefly without detailed explanations. The MTL architectures are divided into two categories as hard parameter sharing and soft parameter sharing (Figure 2).

Hard parameter sharing, which has been around since the early days of Caruana's work [53], involves sharing the hidden layers of a model across all tasks but keeping separate output layers for each task. As in Caruana's work, Long and Wang [60] proposed a Deep Relationship Network which has shared convolutional and task-specific fully connected layers with matrix priors in order for the model to learn the relationship between tasks. Nevertheless, shared convolutional layers were predefined in their design, which may degrade the performance of some tasks. In contrast to hand-designed architectures, Lu [61] suggested a method for automatically learning the architecture of a deep neural network for multiple tasks by using a dynamic branching process. This process makes decisions about which tasks should share features at each layer of the network, considering both the relatedness of the tasks and the complexity of the model.

In contrast to supervision from all tasks at the outermost level, Søgaard and Goldberg [62] suggested that supervision of some low-level tasks in lower levels improves learning of high-level tasks in natural language processing. Another approach based on different levels of sharing knowledge was "a joint many-task model" [63]. Authors argued that this approach allows the network to learn more effectively, as the lower layers can learn features that are more general and applicable to a wide range of tasks, while the higher layers can specialize in tasks that require more specialized and task-specific features. Overall, the proposed approach enabled the MTL model to be able to learn a wide range of Natural

Language Processing (NLP) tasks effectively, with a focus on growing itself dynamically to incorporate new tasks.

In addition to predefined or learned shared layers, Liu [64] introduced a Multi-Task Attention Network that consists of a single shared network that learns shared features across all tasks. For each task, rather than using the shared features directly, the model applied a soft attention mask at each convolution block in the shared network. This attention mask determines the importance of the shared features for the specific task and allows the model to learn both shared and task-specific features in an end-to-end manner. This approach enables the model to learn more expressive combinations of features for generalization across tasks while still allowing for task-specific features to be learned.

Besides methods utilizing hard parameter sharing, there are different techniques that utilize soft parameter sharing, which involves sharing certain parts of the model across tasks but allowing each task to have its own set of parameters as well. This allows each model to learn task-specific features without interfering with the learning of other tasks while still being able to benefit from shared knowledge. In [65], Misra et al. proposed a combination of two separate networks called "cross-stitch networks", which allows the model to learn task-specific features while also sharing information between tasks. The authors proposed using a "cross-stitch unit" to combine the features learned by multiple task-specific subnetworks and allow them to influence each other. The authors also proposed using a "cross-gate" to control the flow of information between tasks, allowing the model to choose which features to share between tasks selectively. Thanks to soft parameter sharing, they also showed that cross-stitch networks could be used to transfer knowledge from a model trained on a large dataset to a model trained on a smaller dataset, improving the performance of the smaller model.

Like cross-stitch networks, Duong et al. [66] presented a method utilizing soft parameter sharing between source and target language model for improving the performance of dependency parsers on low-resource languages (languages with limited available training data). To benefit from cross-lingual parameter sharing, which involves sharing the parameters of a parser trained on a high-resource language with a parser for a low-resource language, the authors introduced a language-independent layer in the network that is shared across all languages. This layer was trained on a high-resource language and then finetuned on the low-resource language. The other layers of the network, which are language-specific, were trained only on the low-resource language. In this way, the low-resource parser can "borrow" the knowledge of the high-resource parser, potentially improving its performance.

While previous methods assisted different networks for each of the tasks with sharing knowledge between each other, Ma et al. [67] propounded a "multi-gate mixture-of-experts" model, which consists of a group of bottom networks, each of which is called an expert and task-specific tower networks on top of expert networks. There is also a gating network for each task, which takes the input features and outputs softmax gates that assemble the experts with different weights. This allows different tasks to utilize the experts differently. The results of the assembled experts are then passed into task-specific tower networks. In this way, Multi-gate Mixture-of-Experts (MMoE) model captures the relationships between tasks by learning different mixture patterns of experts for each task through the gating networks. This allows the model to adaptively balance the contribution of each expert network to the overall prediction based on the input data. As a result, MMoE removes performance drop when task correlation is low in hard parameter sharing and reduces parameters originating from task-specific layers in soft parameter sharing.

In [68], authors utilized MMoE in their multitask ranking system for recommending videos to users. The system was designed to address the problem of limited available data for individual tasks in the recommendation system by leveraging information from multiple related tasks through parameter sharing. They proposed adding expert layers on top of a shared layer which encodes and reduces the dimensionality of the input layer in order to minimize model training and serving costs.

2.2.2 Continual Learning

Biological neural networks have the ability to acquire new knowledge from sequential experiences while extending and preserving their old knowledge. On the other hand, artificial neural networks suffer from catastrophic forgetting, which means a significant decline in performance on previously learned tasks when it is trained on new tasks [7]. This is typically observed in computational models that are trained sequentially on a series of tasks rather than being trained on all tasks concurrently. Although ANNs are very promising in single tasks coming from independent and identically distributed data distributions, they update their parameters remarkably wrt. non-stationary data [69], which results in catastrophic forgetting. On the other hand, continual learning requires a learning system to learn and adapt over time, as it is exposed to a stream of tasks rather than learning a single task in isolation.

In continual learning, it is important for the learning system to be able to transfer knowledge between tasks in order to avoid forgetting what it has learned and to improve performance on new tasks, which are requirements of forward and backward transfer. Backward transfer refers to the ability of a learning system to transfer knowledge gained from learning a new task to improve performance on a previously learned task. On the other hand, forward transfer refers to the ability of a learning system to apply knowledge gained from learning a task to improve performance on a different, new task. Backward transfer can help to prevent forgetting by reinforcing previously learned knowledge, while forward transfer can help the learning system to generalize its knowledge and apply it to new situations [70]. Both backward and forward transfer is important for continual learning because they allow the learning system to build upon its previous experiences and adapt to new tasks more efficiently.

To address the issue of catastrophic forgetting and have the specialty of forward and backward transfer in connectionist networks, various techniques have been proposed that draw inspiration from the mechanisms of learning and memory in biological systems. These methods are divided into three main categories such as regularization, parameter isolation, and replay (memory), as seen in Figure 3. Despite the different taxonomies for CL methods such as [71], we followed a similar taxonomy in [72] for the categorization of CL methods which are not mutually exclusive and might overlap in some cases. In the next sections, we presented different methods for alleviating catastrophic forgetting under three main categorizations briefly, whereas CL methods utilized in experiments were discussed in more detail.

2.2.2.1 Regularization Based Methods

The human brain is able to maintain stable patterns of behavior and thought in order to function effectively in the world while it is also able to adapt and change in response to new experiences and environments. This concept in the brain is known as the stability-plasticity dilemma [73]. One way to conceptualize this dilemma is through using neural network models, which have been used to study the

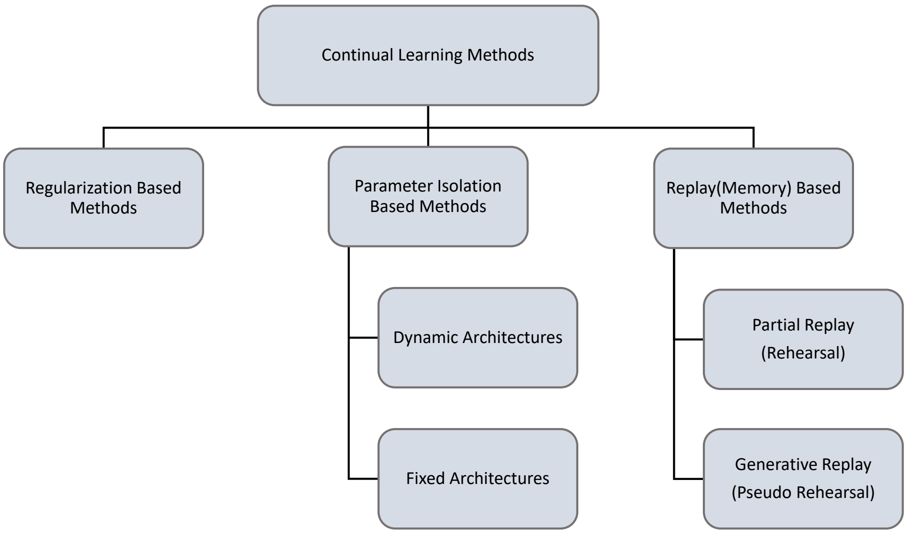

Continual Learning Methods
Regularization Based Methods
Parameter Isolation Based Methods
Replay(Memory) Based Methods
Dynamic Architectures
Fixed Architectures
Partial Replay (Rehearsal)
Generative Replay (Pseudo Rehearsal)

Figure 3: A taxonomy of continual learning methods.

relationship between stability and plasticity in the brain. In these models, stability is often associated with the maintenance of existing neural connections, while plasticity is associated with the formation of new connections or the modification of existing ones.

The stability-plasticity dilemma has been studied in relation to learning and memory in cognitive science. For example, research has shown that the balance between stability and plasticity is important for effectively retaining and updating information in the brain [74]. Stability is necessary for the brain to maintain previously learned information and to prevent interference from new information. On the other hand, plasticity is essential for the brain to update and reorganize its neural connections in response to new experiences and to facilitate learning.

The stability-plasticity dilemma in neuroscience examines the relationship between the development and function of neural networks in the brain. For example, research has shown that the balance between stability and plasticity is important for the development of brain circuits during early life and for the maintenance of these circuits in adulthood [75]. One way that the brain's plasticity can be modified is through the process of long-term potentiation (LTP), which is a form of plasticity that involves an increase in the strength of synapses in response to repetitive or high-frequency stimulation [76]. Metaplasticity [77], a concept that refers to the plasticity of synapses, is believed to regulate the balance and shape the way the brain adapts and changes in response to various stimuli, such as inducing LTP.

A fundamental principle of neural plasticity was proposed by Hebb [78]. Hebb's rule states that neurons that fire together will wire together, meaning that the strength of the connection between two neurons will increase as a result of their repeated co-activation. According to Hebb's rule, when a neuron is repeatedly activated by another neuron, the connection between them is strengthened, leading to more efficient communication between them. This process is thought to underlie the formation of new neural connections during learning and memory formation.

[79] proposed a computational model for memory consolidation that involves the interaction of two different types of memory: fast, labile memory that is stored in the synapses of neurons; and slow, stable

memory that is stored in the structural changes of neurons. According to the model, during the consolidation process, fast memory is gradually transformed into slow memory through the strengthening of certain synapses and the weakening of others. In conjunction with [79], parameters of connectionist models are updated to alleviate catastrophic forgetting by restriction coming from extra regularization terms coming from loss function in regularization-based methods [80].

One of the well-known methods utilizing regularization for alleviating catastrophic forgetting is Learning without Forgetting [81]. In Learning without Forgetting (LwF), the model was trained by only images and labels from the current task. Similar to knowledge distillation [50], which is the method for transferring knowledge from a teacher model to a student model, LwF employed class probabilities of images from current task obtained by the pre-trained model of previous tasks as soft labels and distills knowledge to the new model. In the beginning, output probabilities for each class of previous tasks were obtained from images belonging to new tasks via the original (pre-trained) model. Subsequently, the distillation of the new network with additional outputs originating from the new tasks encouraged the probabilities of class predictions for old tasks to close the previous probabilities. Besides knowledge distillation loss, multi-label loss for new tasks was also included in the total loss. Although LwF has shown promising results when tasks are related, Aljundi et al. [82] suggested that it is sensitive to tasks coming from different distributions.

Another method similar to LwF is the method in [83], which proposed a method for preserving the knowledge of previous tasks while learning a new task by utilizing autoencoders. Encoder Based Lifelong Learning (EBLL) model is composed of three parts which are a feature extractor, shared layers, and task-specific layers, and additional under-complete autoencoders connected to the feature extractor. For each task, an autoencoder was trained in order to map high-level features to low-level features. While LwF employs the distillation of soft labels, EBLL distills knowledge from low-level features, which enabled the model to become less sensitive to the data distributions.

Apart from the methods taking advantage of data [72], there are also methods employing model parameters, one of which is Elastic Weight Consolidation [84], which is a method for addressing catastrophic forgetting in neural networks by adding a penalty term to the objective function of the model during training. The penalty term encourages the model to maintain the weights of the network that are important for the previous tasks while still allowing the model to adapt to the new task. The penalty term is based on the Fisher information matrix, which is a measure of the amount of information that is stored in the weights of the network. The Fisher information matrix is calculated using the second derivatives of the loss function with respect to the weights of the network, and it gives a sense of how sensitive the loss function is to changes in the weights of the network. Elastic Weight Consolidation (EWC) works by encouraging the model to maintain the weights that have a high value in the Fisher information matrix, as these weights are more important for the previous tasks, and changing them would have a larger impact on the loss function.

Synaptic Intelligence (SI) is another method of making use of regularization. In [85], Zenke et al. suggested that intelligent synapses gradually accumulate relevant knowledge for a particular task, and this accumulated knowledge is used to store new information without losing previously learned information quickly. In other words, SI also penalizes changes in the important weights as in EWC [84]. However, it computes the importance of weights online, whereas EWC computes them after the training phase.

Unlike previous approaches, Aljundi et al. [86] suggested a novel approach for lifelong learning called Memory Aware Synapses, which does not rely on labeled data or loss functions to determine the importance of the parameters in a neural network. Instead, it calculates the sensitivity of the output function to changes in each parameter in an unsupervised and online manner. This allows Memory Aware Synapses (MAS) to adapt to specific test conditions and continuously update the importance weights of the network parameters without overwriting important knowledge related to previous tasks. Authors argued that this approach is necessary for preserving knowledge in the face of limited model capacity and an unlimited amount of new information to be learned.

2.2.2.2 Parameter Isolation Based Methods

In the brain, there are different processes for acquiring new knowledge and mitigating catastrophic interference. One of them is neurogenesis, the process of generating new neurons in the brain primarily during development [87]. However, this generation can also occur in certain regions of the adult brain, such as the hippocampus, which is involved in learning and memory [87]. There is evidence that neurogenesis is associated with learning and memory. For example, physical exercise and environmental enrichment, which increase neurogenesis in the hippocampus, can improve learning and memory in animals [88]. In contrast, inhibiting neurogenesis can impair learning and memory, while increasing neurogenesis can enhance these functions [89]. The exact mechanisms by which neurogenesis contributes to learning and memory are not fully understood, but it is thought that new neurons may play a role in the formation of new memories and the integration of new information with existing knowledge [89]. New neurons may also help to preserve old memories by strengthening the connections between neurons, a process called neuroplasticity [89]. Overall, neurogenesis is an important process that may contribute to the brain's ability to acquire new knowledge and preserve old information.

These mechanisms are adapted for connectionist models in association with natural cognitive systems. Similarly, neurogenesis and neuroplasticity are realized by the isolation of parameters in ANNs. One group of parameter isolation-based methods overcome catastrophic forgetting by extending the model if there is no restriction in size. On the contrary, the other group keeps the model size fixed, and it allocates parameters for different tasks by pruning the models.

The first group of methods promoting parameter isolation is built on dynamic architectures. Rusu et al. suggested a method called progressive neural networks for training large, deep neural networks that gradually increase the network's capacity over time [90]. The main idea behind progressive neural networks is to start with a small network and gradually increase its capacity by adding a new network with lateral connections to old networks. In the training stage, all parameters in old networks are frozen, and only the parameters of the network responsible for the current task are updated. Thanks to lateral connection, knowledge from old tasks are transferred to new tasks, which satisfies forward transfer and the retention of old knowledge through frozen networks.

As progressive neural networks have different networks for each task, Aljundi et al. [82] proposed specialized expert gates for each task. The main idea of the "Expert gate" approach is to divide the model into a set of "experts", each of which is responsible for learning a subset of the tasks that the model is expected to perform. These experts are connected through a "gate", an autoencoder, which determines which expert should be used to solve a given task. The gate module is trained to select the most appropriate expert for a given task based on the expertise of each expert and the difficulty of the task. Incorporation of new tasks into the model involves training a new expert to handle the new task and updating the gate module to select the appropriate expert for each task. Both progressive neural networks and expert gate increase model size significantly and in the same amount each time due to the addition of a new network for each new task instead of a small number of neurons.

On the other hand, Yoon et al. [91] proposed Dynamically Expandable Network (DEN), which grows with the addition of a variable number of neurons. According to [91], DEN involves three components which are selective retraining, dynamic network expansion, and network split or duplication, respectively. Firstly, Yoon et al. identified neurons that are relevant to the new task and selectively retrained the network parameters associated with them. Subsequently, if the selective retraining fails to obtain the desired loss below a set threshold, the network capacity is expanded in a top-down manner, while unnecessary neurons are eliminated using group-sparsity regularization. In addition, the authors used a technique called "network split/duplication" to identify neurons that have drifted too much from their original values during training and duplicate them to stabilize the weights of the network and prevent them from changing too much as new tasks are learned.

Similar to DEN, Xu and Zhu [92] suggested a method for expanding a CNN-based task network with the help of an Long Short Term Memory (LSTM) network which controls the number of filters and nodes to be added to the task network. Similarly, Learn to Grow (LtG) [93] involved explicitly separating the learning of model structures and the estimation of model parameters. This means that the model's structure (e.g., the number and arrangement of layers and neurons) is optimized for each task independently of the specific parameter values (e.g., the weights and biases of the connections between neurons). The structure of the model was found through an architecture search process, which considers various options, such as reusing previous layers or introducing new ones. Once the optimal structure has been identified, the model parameters are then estimated based on that structure.

Another method of altering the structure of the model is Packing and Expanding (PAE) [94] that was built on the approaches used in ProgressiveNet [90] and PackNet [94]. ProgressiveNet avoids catastrophic forgetting by reusing the weights learned for previous tasks, but this can result in a redundant structure. PackNet avoids forgetting by compressing the deep model through weight pruning and retraining the remaining weights, but it does not allow for the extension of the model architecture. PAE addresses these limitations by using an iterative pruning procedure to compress the model and selectively expanding the architecture by adding filters. The old-task weights are re-used and remain fixed, and additional weights are added from the previously saved ones by repeating the iterative pruning process. If the desired accuracy has not been achieved, the architecture can be further expanded and the process repeated. Experiments were conducted on face verification, gender recognition, and age estimation tasks.

Similar to PAE, Hung et al. [95] applied compacting and growing techniques for new tasks. Furthermore, Compacting, Picking, and Growing (CPG) has a picking step that selects useful weights belonging to old tasks using a learnable mask, whereas PAE makes use of whole weights of old tasks. In CPG, compacting aims to reduce the size of the model's parameter space by identifying and removing unnecessary parameters that are not relevant to the current task. This is done by computing the importance of each parameter based on its contribution to the performance of the current task and pruning the low-importance parameters. Picking is designed to identify a subset of the most significant weights from the old tasks and reuse them for the forward transfer of old knowledge. Finally, growing is intended to expand the model's capacity when necessary by adding new parameters to the model. This is done by monitoring the model's performance on the current task and adding new parameters when the model's performance starts to degrade.

Along with dynamic architectures, there are also methods promoting fixed architectures for mitigating catastrophic forgetting. However, the performance of the later tasks will degrade due to a lack of parameter allocation when the number of tasks increases in this type of method. Fernando et al. suggested a method called PathNet [96], which has different modules for learning tasks. They used a type of evolutionary computation called "genetic algorithms" to optimize the structure of the network. They did this by representing the structure of the network as a "genome," which consists of a series of "paths" through the network. These paths are used to connect the input and output layers of the network and can be modified during training by the genetic algorithm.

Similarly, Serrà et al. [97] suggested a mechanism called Hard Attention to the Task (HAT), which is a task-based attention mechanism that allows a model to retain information from previous tasks while learning a new task. HAT does this by learning almost-binary attention vectors through gated task embedding and using the attention vectors of previous tasks to create a mask that constrains the updates of the model's weights on the current task. The mask is almost binary, meaning that some weights remain unchanged while the rest are adapted to the new task.

There is also a gating mechanism that allows the creation and destruction of paths across layers that can be later preserved when learning a new task, similar to the approach used in the PathNet algorithm. However, unlike PathNet, the paths in HAT are not based on modules but on individual units, allowing the network to learn and automatically dimension paths for individual units and ultimately affect the weights of individual layers. Additionally, unlike the PathNet approach, which uses genetic algorithms to learn paths in a separate stage, HAT learns the paths along with the rest of the network using backpropagation and stochastic gradient descent.

In contrast to previous methods promoting freezing some parameters from old tasks, Mallya and Lazebnik [98] proposed a method for adding multiple tasks to a single neural network by iterative pruning the network after training the entire network. The idea is to, with a pre-trained network that has been trained on a single task, identify the most important layers for the original task, prune the non-critical layers by removing some weights and biases, retrain the pruned network on the original task to ensure performance is not degraded, and repeat this process for each additional task added to the network.

2.2.2.3 Replay or Memory Based Methods

Memory is a fundamental cognitive function that allows us to encode, store, and retrieve information about the world around us. Memory is essential for our daily functioning, as it enables us to learn from our experiences and adapt to new situations. There are many different forms of memory, including short-term memory, which allows us to hold onto information for brief periods of time, and long-term memory, which enables us to retain information over longer periods of time.

The brain has a complex and interconnected system for learning and memory, and a significant amount of research has been devoted to understanding how this system works. One influential theory that has been proposed to explain the brain's learning and memory system is the Complementary Learning Systems (CLS) theory.

The Complementary Learning Systems theory proposes that the brain contains multiple systems for learning and memory and that these systems work together to support various forms of learning and

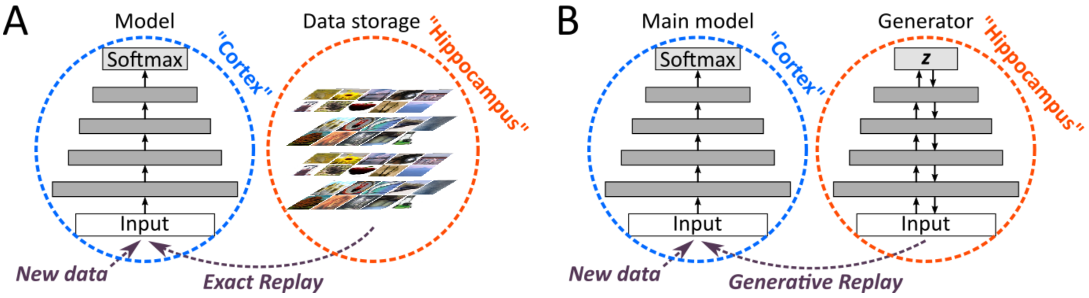


Figure 4: Mapping of addition of replay in ANNs to the human brain. (A) Exact replay which analyzes the hippocampus as a buffer for storing memories or episodes. (B) Generative replay (pseudo rehearsal) which analyzes the hippocampus as a separate generative neural model. The figure was taken from [4].

memory [99]. One key component of the CLS theory is the distinction between the hippocampus, which is thought to play a critical role in the formation of new memories, and the neocortex, which is thought to be involved in more permanent forms of learning. According to the CLS theory, the hippocampus and the neocortex work together to support learning, with the hippocampus providing a temporary storage system for new information and the neocortex gradually integrating this information into more permanent memories. The concept of "reactivation" is also an important aspect of the CLS theory, as it refers to the process by which memories are strengthened and consolidated over time through retrieval and integration into more permanent networks of neurons in the neocortex.

There is a significant body of research that supports the CLS theory. For example, studies have shown that the hippocampus is necessary for the formation of new memories and that the neocortex plays a critical role in the consolidation of these memories over time [100]. Other research has also demonstrated the importance of reactivation in the consolidation of memories [101]. The CLS theory was also proposed in part as a response to the limitations of connectionist models of learning and memory, which have difficulty accounting for the complex patterns of forgetting and remembering that are observed in humans, and other animals [102].

The case of H.M., a patient who underwent surgery to remove large portions of his hippocampus and other brain structures in an effort to treat epilepsy, has provided important insights into the role of the hippocampus in long-term memory formation. H.M.'s surgery resulted in severe impairment in his ability to form new long-term memories but did not affect his short-term memory or his ability to remember information from before the surgery. This finding has been interpreted as strong evidence for the role of the hippocampus in the formation of new long-term memories and has helped to support the CLS theory [103]. In addition to its importance in the CLS theory, H.M.'s case has also been used to help identify the neural basis of long-term memory formation and to inform the development of treatments for memory impairments. Overall, the CLS theory provides a useful framework for understanding how the brain supports learning and memory. Deriving from replay in biological networks, replay(memory) based methods are proposed in connectionist networks.

Replay-based methods can be discussed in two categories: partial replay(rehearsal) and Generative Replay (GR) (pseudo rehearsal) [104] [72]. Figure 4 visualizes how different replay-based models take inspiration from the brain, especially the hippocampus. Figure 4A indicates that the hippocampus is viewed as a memory buffer, and ANNs learn from the replay of the exact samples like humans. On the other hand, Figure 4B demonstrates how the hippocampus is viewed as a generative model and mapped to generative ANNs.

One of the partial replay methods is Incremental Classifier and Representation Learning (iCaRL) [105] which is a method for incrementally learning a classifier and a compact representation of the data, allowing it to continuously learn new classes without forgetting the ones it has previously learned. iCaRL method begins by training a classifier and representation on a set of initial classes. When a new class is introduced, iCaRL predicts the class of a small number of example images (exemplars) from that class using the classifier. Then, the exemplars are used to replay the old classes to the classifier, strengthening the connections which are used to classify them during training.

In order to update the representation for the new class, iCaRL calculates the mean feature vector of the exemplars and adds it to the representation. Then, the classifier is updated by using both the exemplars and the updated representation. When iCaRL classifies a new example image, the feature vector of the example is computed initially using the updated representation. Then, iCaRL uses a nearest-mean-of-exemplars classification technique, which involves measuring the distances between the feature vector and the mean feature vectors of the learned classes, to predict the class.

Another method that focuses on sample selection is [106]. Aljundi et al. presented a method for improving the performance of continual learning algorithms by using constraint optimization to guide the selection of training examples. They formulate this selection process as a solid angle minimization problem and propose a surrogate objective to solve it. To further improve sample selection for large datasets, the authors propose a greedy algorithm that is efficient and resistant to imbalanced data streams. Lopez-Paz and Ranzato proposed [107] GEM, which projects the current task gradient into a feasible region defined by the gradients of previous tasks by solving a constrained optimization problem. These updates are constrained in such a way that they do not increase the loss of previous tasks.

Although several other methods concentrated on the selection of samples, such as [108], there are several more cognitively plausible methods that replay internal(hidden) representations rather than raw pixels. Hayes et al.[109] proposed the REMIND, a brain-inspired method, which uses tensor quantization to store and efficiently retrieve hidden representations (such as feature maps) for replay in order to mitigate forgetting in Convolutional Neural Networks (CNNs). This is achieved through the use of Product Quantization [110]. Correspondingly, Pellegrini et al. [111] presented a technique called latent replay that involves storing activations (intermediate outputs) at a specific layer in a neural network, rather than storing raw data inputs, in order to reduce computation and storage requirements. To ensure that the representation remains stable and the stored activations are still relevant, authors suggest slowing down learning at layers below the one where the activations are being stored while allowing learning to continue for layers above it.

Alternative to partial replay methods, which raise privacy concerns in addition to the storage problem, methods utilizing generative replay (pseudo rehearsal) are much more biologically plausible due to the need for storage of raw pixels in biological neural networks. In [112], Shin et al. proposed the Deep Generative Replay (DGR), which generates samples for old tasks and employs them by combining the images from the current task in the training of the classification model. Van de Ven and Tolias [113] advanced GR by the addition of feedback connection. While DGR [112] uses two models, one for GR and one for the task solver, the Replay-through-Feedback method merges GR into the main task solver by attaching backward connections. Moreover, the authors suggested knowledge distillation of soft labels in training. Despite the different generative models, Lesort et al. [114] concluded that original GR is better than others in continual learning from their experiments.

Furthermore, there are also more cognitively inspired methods, one of which is FearNet [115], designed to be memory efficient by using a dual-memory system. This system consists of a network for recent memories inspired by the hippocampus, and a network for long-term storage, inspired by the medial prefrontal cortex. The model also includes a module inspired by the basolateral amygdala that determines which memory system to use for recall.

Similarly, van de Ven et al. [4] suggested a method that is inspired by the way the brain processes and retains information and involves several modifications to the standard approach to continual learning using generative replay. One modification is the merging of the generator, which is responsible for generating replay, into the main model by adding generative feedback connections. This allows the model to correspond more closely to the hierarchical structure of the brain, with the first few layers corresponding to the early layers of the visual cortex and the top layers corresponding to the hippocampus. Another modification is the use of a Gaussian mixture prior over the latent variables in the model's Variational AutoEncoder (VAE), allowing the model to generate specific classes by restricting the sampling of the latent variables to their corresponding modes. To achieve context-dependent processing, the decoder part of the network is conditioned on an internal context representing the specific task or class to be generated or reconstructed. The model also replays previously learned classes internally, at the hidden level, rather than all the way to the input level, in order to mimic the way the brain processes information closely.

2.2.3 Related Work on Face Analysis

Face detection is a computer vision task that involves identifying the presence of human faces in images or videos. FD is a crucial step in many applications, including security systems, face recognition, and human-computer interaction. One of the challenges in face detection is that faces can vary significantly in terms of size, orientation, and appearance due to factors such as lighting, facial expressions, and facial attributes. To address these challenges, face detection algorithms often incorporate techniques such as scale invariance, robust feature extraction, and cascade classifiers.

One of the earliest works that studied face detection in a cascaded fashion in order to boost the performance of face detection in uncontrolled environments was [116]. Li et al. proposed a CNN cascade, which consists of six cascaded CNNs including three for binary classification (face or non-face) and three for calibration of bounding boxes. In their design, an image is scanned using the 12-net at various scales to eliminate a large number of detection windows quickly. The remaining windows are processed by the 12-net to adjust their size and location. Non-maximum suppression is then applied to remove highly overlapped windows. The remaining windows are resized to 24x24 and processed by the 24-net, which eliminates more windows, then adjusted by the 24-net and subjected to non-maximum suppression again. The final step involves the 48-net evaluating the remaining detection windows, which are then calibrated by the 48-net and output as residual detection bounding boxes after non-maximum suppression was applied using an Intersection-Over-Union threshold.

Like [116], Multitask Cascaded Convolutional Networks [117] utilizes three cascaded convolutional networks for joint face detection and facial landmark detection. Zhang et al. used the Proposal network for collecting candidate faces in the first stage. After Non Maximum Suppression (NMS) is applied to candidates, the remaining windows are passed through the Refine network and NMS for rejecting false candidates and calibration of bounding boxes, respectively. In the final stage, the remaining bounding boxes are fed to the Output network, which produces five facial landmarks' coordinates, including the corners of the mouth, the tip of the nose, and the center of the eyes, in addition to bounding boxes. Zhang and Zhang [118] proposed another method that employs cascaded networks for FD and FL with an additional task, face pose estimation. They utilized a boosting-based multiview face detector [118] for obtaining image patches in the first step. After preprocessing image patches with histogram equalization, linear lighting removal, and intensity normalization, their multitask DCNN produces predictions for face/nonface, face pose, and facial landmarks. While Multitask Cascaded Convolutional Neural Networks (MTCNN) trains three networks for leveraging the inherent correlation between FD and FL, DCNN is applied for filtering predictions from a multiview detector.

As in DCNN, Zhang et al. [119] presented a multitask model called Tasks-Constrained Deep Convolutional Network, which accepts cropped face images for facial landmark detection and pose estimation. They trained TCDCN to jointly optimize facial landmark detection together with other tasks that are correlated with facial landmark detection, such as head pose estimation and facial attribute inference such as gender, smiling, and glasses detection for improving the robustness of facial landmark detection, especially in the presence of occlusion and pose variation. Zhang et al. also proposed a task-wise early stopping method to facilitate learning convergence. Another method that employs multitask learning for face detection, landmarks localization, pose estimation, and gender recognition is HyperFace [120], which consists of three modules. In the first module, Region-based CNN [121] generates region-proposals via Selective Search algorithm [122] from images and scales them to 227x227 pixels for the second module. In the second module, an AlexNet [9] based model performs classification for face detection, localization of 21 landmarks, visibility factor for landmarks, pose estimation(row, pitch, yaw), and gender recognition by fusing the intermediate layers of the network. The final module processes predictions for improving the performance of tasks by using Iterative Region Proposals and Landmarks-based NMS. They also propounded a model based on ResNet-101 [123].

All previous methods followed the same procedure: detecting faces with one network and then applying another network to detected patches for estimating other tasks. However, multi-stage networks become inefficient when the number of detected faces in the first stage increases due to the second stage. There are also methods that predict face bounding boxes and landmark locations together in single-stage. One of them is RetinaFace [124], which performs face detection, pixel-wise face localization, and pixel-wise 3D shape face information. Deng et al. propounded that joint optimization of face detection and facial landmark detection boost the performance of hard face detection. They employed supervised learning for FD and FL branches and self-supervised learning for the mesh decoder branch for training their Single Shot Detector [125] based multitask model. Anchor boxes were utilized for regressing bounding boxes in Retinaface. Similarly, YOLOv5Face [126] is another method that predicts both bounding boxes and landmark points in one shot using anchors. In addition to methods availing anchors, there are also anchor-free methods like Centeface [127]. Authors modified Centernet [128] in order to make the model faster and more accurate for face detection.

Besides MTL methods utilized for face analysis, there are also methods employing continuous learning for the analysis of faces. Barros et al. [129] proposed a method called the personalized affective memory model for understanding person-specific facial expressions. Their proposed model consists of two sub-modules which are the prior-knowledge learning module and the affective memory module. The former utilizes an adversarial autoencoder, which encodes information about facial expressions and generates faces with different expressions for a person. Subsequently, their second module, the growing-when-required model, produces representations for generated faces of a person with different expressions and stores them in the cluster of personalized affective memories. Finally, the emotion of a person is recognized by the utilization of that cluster. Their method utilizes unsupervised clustering settings, which facilitates the learning of new person-specific emotions continuously.

Similarly, Churamania and Gunes [130] suggested a method called a Continual Learning Framework with Imagination for Facial Expression Recognition (CLIFER). They utilized the Growing Dual Memory, which includes episodic memory and semantic memory modules. As the theory of Complementary Learning System states, episodic memory is a fast-learning mechanism for non-overlapping representation, whereas semantic memory accumulates knowledge by overlapping representations slowly. They also employed an imagination model in order to generate different expressions for a person, which are used for training the dual-mechanism model.

Additionally, there is a method employing the generation of internal representation instead of raw images. Mainsant et al. [131] propounded Dream Net that involves two fully connected layers, a learning net, and a memory net. Their models accept encoded features obtained via a ResNet50 model trained on a facial expression recognition dataset. The learning net takes the extracted representation and regenerates the input representation along with classification probabilities. After training the learning net, its parameters are copied to the memory net, which generates input representations and classification labels from random noise.

Although previous works focused on the utilization of CL methods for facial expression recognition, Hung et al. introduced CPG [95], which comprises three steps. The first step prunes the model gradually after training the model for a task. In the second step, the model and a learnable binary mask for picking weights belonging to old tasks are trained for the current task. Finally, the model is expanded if the target performance can not be reached. In their experiments, they tested their model on facial-informatic tasks (face verification, gender, expression and age recognition) in addition to classification tasks. The face images were detected and aligned with the help of MTCNN algorithm, and all face analysis tasks were learned sequentially.

CHAPTER 3

MULTITASK LEARNING FOR FACE DETECTION AND FACIAL LANDMARK DETECTION

In multitask learning, a machine learning model is trained to perform multiple tasks simultaneously. This is inspired by the way that the human brain is able to multitask by performing multiple cognitive tasks at the same time, such as listening to music while driving a car. In the field of machine learning, multitask learning is often used to improve the performance of a model on a particular task by training it on a related set of tasks. For example, a model that is trained to classify images of different animals may perform better if it is also trained to classify images of plants and objects. This is because the model can learn shared features and patterns that are useful for both tasks. Overall, the idea of multitask learning in machine learning is inspired by the way that the human brain is able to multitask and perform multiple cognitive tasks simultaneously.

In this chapter, we analyzed the effect of multitask learning on two related tasks, which are face detection and facial landmark detection. Face detection is the process of locating and extracting the face region from images or videos. Many deep learning-based CNN algorithms, such as [117] [124] [126], were designed for FD. They generally produce a confidence score which is how probably the region contains a face and coordinates for face regions like centers, width, and height of faces. On the other hand, facial landmark detection is the process of automatically identifying and localizing specific points of interest on the face, such as the corners of the mouth, the center of the eyes, and the tip of the nose. These landmarks are helpful for tasks such as face recognition, facial expression analysis, and head pose estimation. Firstly, we introduced our proposed connectionist network for FD and FL. Later, we continued with our experiments in order to examine how multitask learning influences the performance of FD while learning another task, FL.

3.1 Proposed Approach

In this section, we presented our multitask model, which is based on YOLOv5 [5] object detector. YOLO [132] has different versions, each of which boosts the performance of previous models. In YOLO, authors designed a real-time single-stage object detector that predicts bounding boxes and class probabilities at the same time. It is actually a multitask network that solves object detection using regression instead of classification. The key idea behind YOLO is to use a convolutional neural network to predict the bounding boxes and class probabilities of objects in an image. Given an input image, the CNN first processes the image through a series of convolutional and max pooling layers

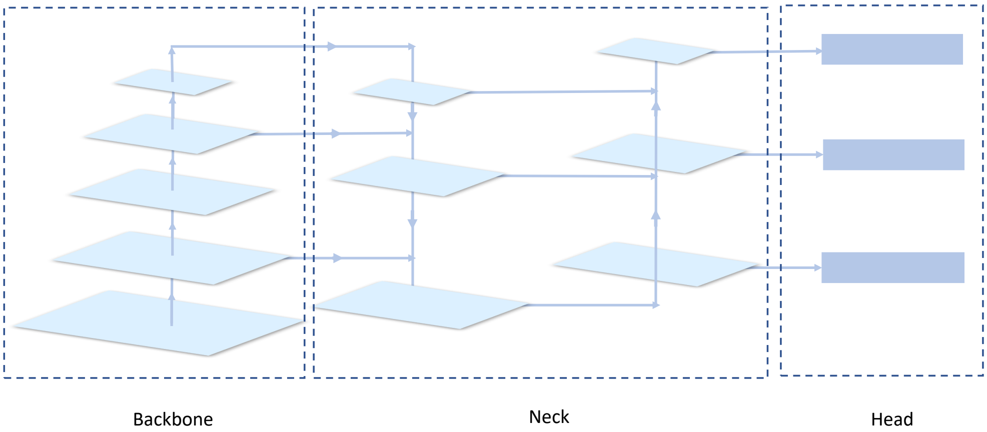


Figure 5: General architecture of YOLOv5 [5] model.

to extract features from the image. These features are then passed through a series of fully connected layers, which predict the bounding boxes and class probabilities of objects in the image.

One of the key advantages of YOLO is its ability to handle multiple object classes and scales. It uses a grid-based approach, where the input image is divided into a grid of cells, and each cell is responsible for predicting the bounding boxes and class probabilities of the objects that fall within that cell. This allows YOLO to accurately detect objects of different sizes and classes in the same image.

Later, the authors improved their design and renamed their new model YOLOv2 [133]. Batch normalization [134] was added to convolutional layers while the backbone was changed to Darknet-19 [133], and the input size of the models was increased from 224x224 to 416x416 pixels. Furthermore, they used feature maps from different scales and anchor boxes, which were chosen by k-means clustering. In training, the input size of the models was changed to different sizes, and output prediction for classes was modified according to multi-label prediction.

Following YOLOv2 [133], authors improved their design by replacing Darknet-19 with a new backbone Darknet-53 [135] and obtaining predictions from three scales, and they called their new model YOLOv3 [135].

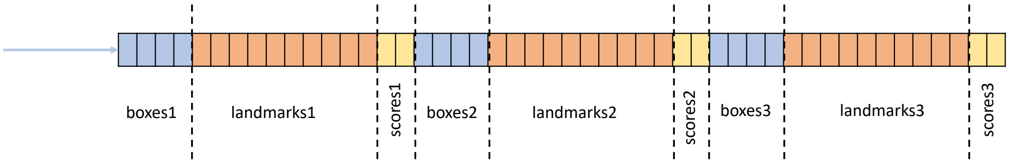


Figure 6: Predictions from the head of our proposed architecture.

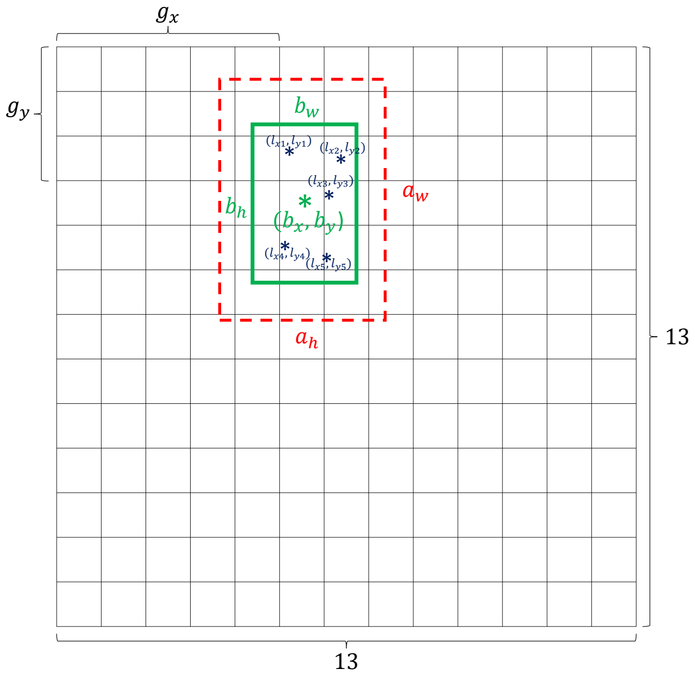


Figure 7: Visualization of predictions for bounding boxes and landmarks on 13x13 grids.

On the other hand, YOLOv5 [5] was designed by a different research team. They replaced the backbone with Cross Stage Partial Network (CSPNet) [136], added Spatial Pyramid Pooling (SPP) [137] and PAN [138] for gathering the features, as in [139]. As shown in Figure 5, YOLOv5 is composed of three parts, which are the backbone, neck, and head. Although all previous YOLO models predict only classes and bounding boxes, we modified the head of YOLOv5 [5] for the detection of five landmark points which are the center of the eyes, the corners of the mouth, and the tip of the nose as in YOLOv5Face [140]. Figure 6 indicates predictions of the modified model. Landmarks have ten predictions due to two coordinates, x and y, for each of the five landmarks. Although YOLOv5Face [140] has several modifications in order to improve the performance of FD and FL, we left YOLOv5 [5] architecture unchanged except for modification of the head for landmark detection. In addition, our model takes inputs with 416x416 pixels size in contrast to 640x640 pixels in YOLOv5 and YOLOv5Face due to faster inference.

In Figure 7, predictions for bounding boxes and landmark points are visualized on 13x13 grids. YOLOv5 produces predictions at three scales, which are the 8th, 16th, and 32th of the input size. Therefore, output feature maps have grids of sizes 52x52, 26x26, and 13x13 in our design due to the input sizes of 416x416 pixels. Bounding box coordinates for each face are obtained from Equation 1. b_x and b_y indicate the coordinates of the center of the bounding boxes while b_w and b_h refer to the width and height of faces, respectively. The center of bounding boxes is calculated from the sigmoid of predictions p_x and p_y. First, the results are multiplied by 2, and then 0.5 is subtracted, which produces offset values between *-0.5* and *1.5* in order to obtain *0* and *1* conveniently [5]. Later, grid distance from the start of grids is added. On the other hand, the width and height of bounding boxes are obtained from the sigmoid of p_w and p_h, which are multiplied by 2, and the result is squared, respectively. Afterward, the results are multiplied by the width and height of predefined anchor boxes.

$$\begin{aligned}
b_x &= (2 * \sigma(p_x) - 0.5) + g_x \\
b_y &= (2 * \sigma(p_y) - 0.5) + g_y \\
b_w &= a_w * (2 * \sigma(p_w))^2 \\
b_h &= a_h * (2 * \sigma(p_h))^2
\end{aligned} \tag{1}$$

Similarly, we obtained landmark predictions from Equation 2. Here, *i* changes between *1* and *5*, which indicates a different landmark. Coordinates of a landmark (l_x^i, l_y^i) are calculated from predictions of the network, (p_{lx}^i, p_{ly}^i), by multiplication of width and height of anchor boxes and addition of the grid distances, (g_x, g_y), respectively.

$$\begin{aligned}
l_x^i &= (p_{lx}^i * a_w) + g_x \\
l_y^i &= (p_{ly}^i * a_h) + g_y
\end{aligned} \tag{2}$$

In training, we utilized multitask loss in Equation 3. The total loss (L_{total}) includes Binary Cross Entropy loss for objectness (L_{obj}) and classification (L_{cls}), complete intersection over union (IOU) loss for bounding boxes (L_{bbox}) and Wing-loss [141] for landmarks (L_{lands}), and $\lambda_{cls,obj,bbox,lands}$ are employed for adjusting effects of each loss in joint optimization.

$$L_{total} = \lambda_{cls} * L_{cls} + \lambda_{obj} * L_{obj} + \lambda_{bbox} * L_{bbox} + \lambda_{lands} * L_{lands} \tag{3}$$

3.2 Experiments

In this section, experiments for comparing single-task and multitask designs on face detection and landmark extraction tasks are described. Firstly, we start by introducing our experimental setup. Afterward, we introduce datasets utilized in training and testing. Finally, experimental results are presented.

3.2.1 Experimental Setup

In experiments, we analyzed the effects of multitask learning on the performance of face detection over single-task learning. Furthermore, joint optimization of tasks was compared with separate optimization of tasks. As a result, we designed four experiments which are given below, in order to develop a multitask YOLOv5 model for FD and FL. All experiments were performed using the same equipment, listed in Table 1.

Table 1: List of equipment used in experiments

Equipment		Version	Details
Hardware	Supermicro Computer	SYS-7048GR-TR	Used for training and testing models
	Processor	Intel Xeon E5-2687W v4 @ 3.00GHz	Processor utilized on the computer
	GPU	Nvidia Quadro P6000 24 GB	GPU utilized on the computer
Software	Operating System (OS)	Ubuntu 16.04 LTS	OS running on the computer
	Nvidia Cuda Toolkit	Nvidia Cuda 11.0	Utilized for accelerated training and testing
	Programming Language	Python 3.6.13	Utilized for designing experiments and calculating results
	Machine Learning Framework	Pytorch 1.8.1	Utilized for designing and evaluating ANNs

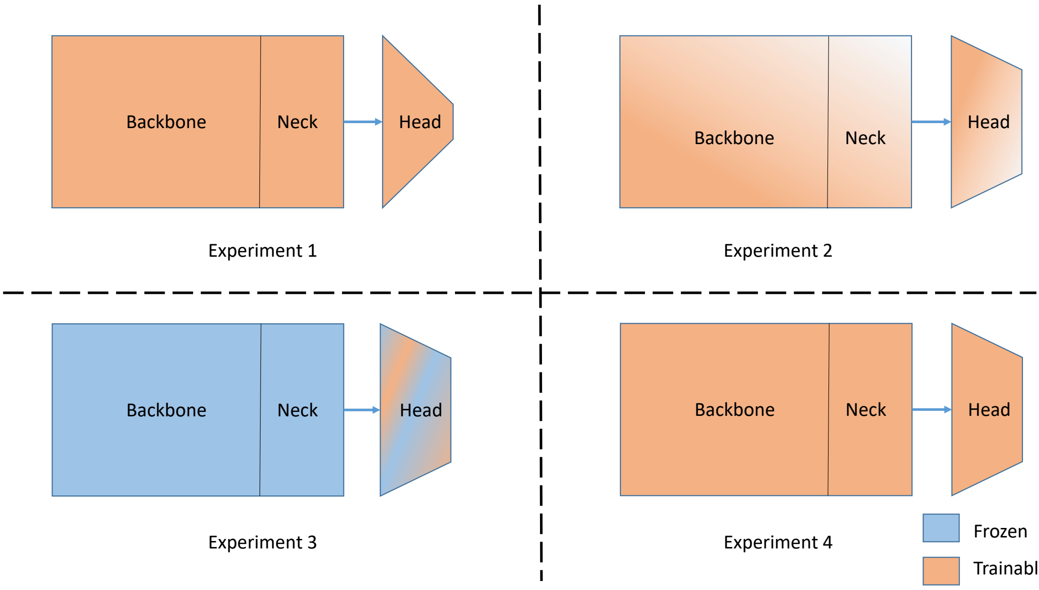


Figure 8: Experimental setups for analyzing the effects of MTL on FD and FL tasks.

Experiment 1. In the first experiment, we trained a YOLOv5 model for only FD task (Figure 8). We initialized the parameters of the model randomly. This setup is like a single-task training using supervised learning despite the fact that both face/non-face classification and bounding box regression are learned together. In the total loss function, we set λ_1 to 0.5, λ_2 to 1.0, λ_3 to 0.05. In this experiment, the FL task was not learned and λ_4 is not used in the total loss function.

Experiment 2. In the second experiment, we trained our YOLOv5 multitask model for FL task. We initialized the parameters of the model from the pre-trained YOLOv5 model for FD task from the first experiment. Figure 8 shows the model with some light parts, which indicates some parameters initialized from FD model. When optimizing the total loss, λ_{1-3} were set to 0.0 whereas λ_4 was set to 0.005.

Experiment 3. In the third experiment, we trained our YOLOv5 multitask model for FL task as in the second experiment. On the other hand, we initialized the parameters of the model from the pre-trained YOLOv5 multitask model for FD task in the first experiment and froze all parameters except parameters only belonging to FL task as Figure 8 indicates. We used the same lambda values λ_{1-4} as in Experiment 2.

Experiment 4. In the fourth experiment, we trained our YOLOv5 multitask model for FD and Fl tasks jointly (Figure 8). We randomly initialized the parameters of the model and set λ_{1-3} as in Experiment 1 and λ_4 as in Experiments 2 and 3.

In all experiments, we trained the models for 500 epochs using Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.937 and applied horizontal flip with 0.5 probability, and mosaic [142] augmentation and hue-saturation-value distortions. Furthermore, blur, median blur, conversion to gray, jpeg compression, and clahe augmentations were applied from albumentations [143] library. The initial

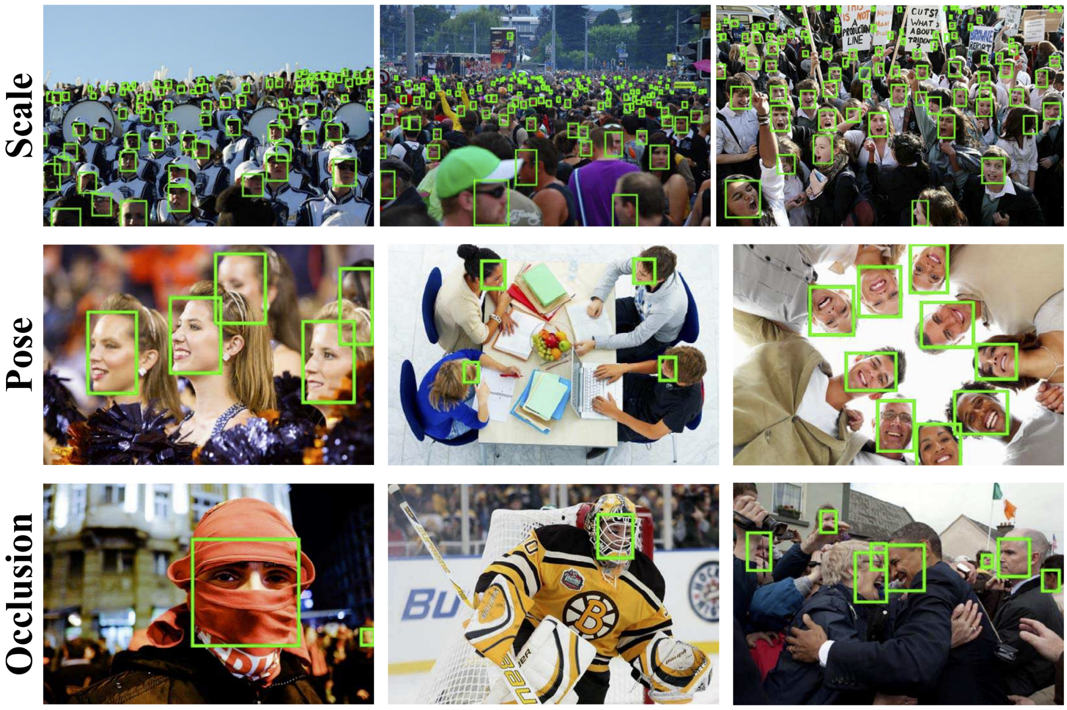


Figure 9: Sample images from Widerface [1] dataset. The image is taken from [1].

learning rate started at 0.01 and decayed until about 0.0001 using YOLOv5's custom learning rate scheduler.

3.2.2 Utilized Datasets

Widerface Dataset

WiderFace [1] is a large-scale face detection dataset, which consists of 32,203 images and 393,703 annotated faces. The dataset is primarily intended for use in the development of facial detection algorithms and is widely used for benchmarking the performance of these algorithms.

The images in the WiderFace dataset come from Wider [144] dataset and span a wide range of visual variations, such as different poses, illuminations, and facial expressions (Figure 9). The images are annotated with bounding boxes that enclose the regions of the face and labels for occlusions, poses, and event categories. The WiderFace dataset is organized into three subsets: the training set, the validation set, and the test set. The training set consists of 12,995 images and 171,542 annotated faces, the validation set consists of 3,000 images and 37,967 annotated faces, and the test set consists of 16,208 images and 183,194 annotated faces. The Widerface dataset is also arranged as easy, medium, and hard sets, which are based on the detection rate of EdgeBox [145] detector.

While the original Widerface dataset has no labels for facial landmark detection, Denget al. [124] provided another version of labels with five facial landmarks annotations (center of eyes, mouth corners, and tip of nose) for a total of 84.6k faces from the training set and 18.5k faces from the validation set along with original labels from the dataset.

In experiments, we took advantage of 12,859 images from the training set of the Widerface dataset for training FD and FL, whereas the validation set of the dataset was employed for measuring the performance of our trained models for only FD task due to the unavailability of landmarks labels for the validation set.

Annotated Facial Landmarks in the Wild (AFLW) Dataset

The AFLW dataset is a collection of annotated images of human faces that have been gathered from the internet. The dataset includes a total of 25,993 faces from around 21,000 images, each of which has been annotated with up to 21 facial landmarks, including points on the eyes, nose, mouth, and jawline (Figure 10). These annotations provide a rich source of information that can be used to train and evaluate machine learning models for tasks such as facial analysis, multi-view face detection, and head pose estimation.

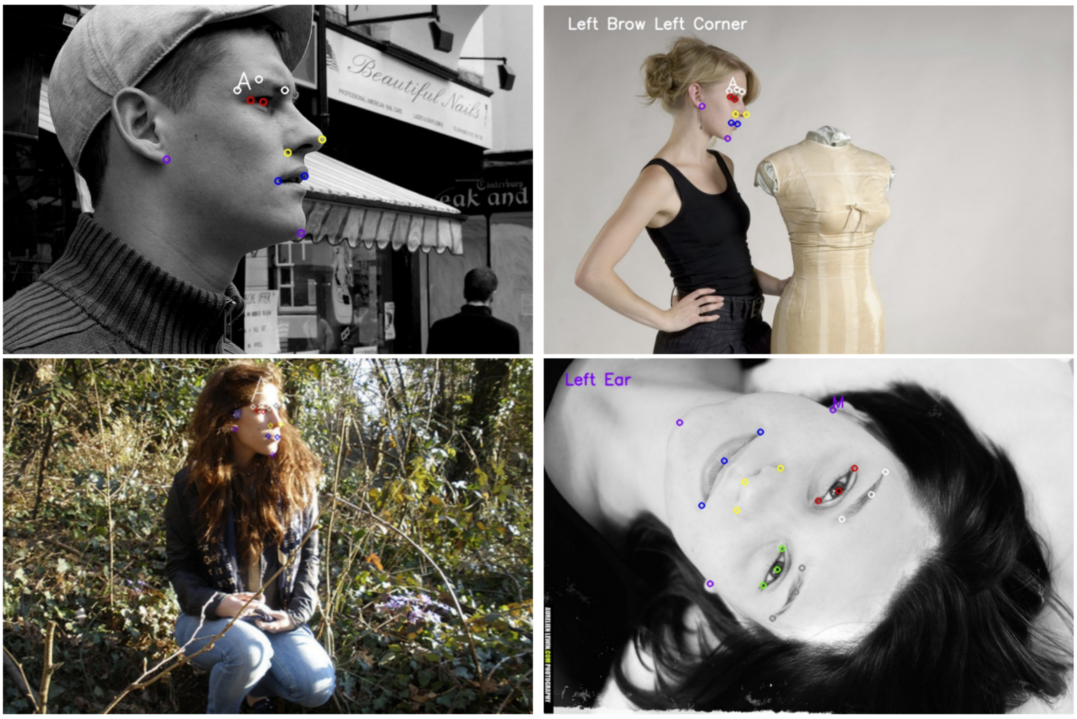


Figure 10: Sample images from AFLW [2] dataset. The image is taken from [2].

One of the key features of the AFLW [2] dataset is its large size and diversity, which make it well-suited for training and evaluating models on a wide range of facial appearances and configurations. The images in the dataset depict a wide range of ethnicities, ages, and genders. The images are also captured in a variety of lighting conditions and poses, providing a more realistic and challenging testbed for facial analysis and recognition algorithms.

Overall, the AFLW [2] dataset is a valuable resource for us in order to evaluate our models for FL task.

Table 2: Comparison of face detection performances of different models on Widerface [1] (easy, medium, hard) validation subset

Model	**AP**		
	Easy	**Medium**	**Hard**
model_base(FD)	0.920	0.888	0.667
model_finetune(FD_FL)	0.860	0.826	0.548
model_freeze(FD_FL)	0.920	0.888	0.667
model_joint(FD_FL)	**0.925**	**0.896**	**0.672**

Table 3: Comparison of facial landmark detection performances of different MTL models on AFLW [2] dataset

Model	**NRMSE**
model_finetune(FD_FL)	0.154
model_freeze(FD_FL)	0.189
model_joint(FD_FL)	**0.041**

3.2.3 Experimental Results

In Table 2, face detection performances of four models were measured on easy, medium, and hard subsets of the Widerface [1] validation dataset. Average precision was used as a performance measurement metric. The highest APs for all three subsets were obtained from the MTL model, which is model_joint(FD_FL). It outperformed the single-task model, which is model_base(FD), on all three subsets. Model_freeze(FD_FL), which utilized frozen weights of model_base(FD), had the same APs with model_base(FD) whereas face detection performance of model_finetune(FD_FL) degraded substantially due to fine-tuning facial landmark task(task2) despite of starting from weights of task1(FD).

In order to better analyze the face detection performances of the models, predictions for face coordinates were given in Figure 11, 12 and 13. Visual results were obtained on the Widerface validation set, which includes challenging examples for the face detection task. Although all models performed the detection of most frontal and medium/large faces, model_joint was generally able to detect occluded or blurry faces better than other models.

The facial landmark detection performances of MTL models are shown in Table 3. Performances were measured via root mean squared error by normalizing interocular distance, which is the distance between the centers of the eyes, as indicated in Equation 4, where l_i and $\hat{l}_i$ are ground truth and predicted landmark point for the location i; n is the total number of landmarks; l_0 and l_1 are the centers of right and left eyes, respectively. The lowest error was produced by model_joint(FD_FL), MTL model optimized tasks jointly. Model_freeze(FD_FL) performed the worst in the FL task with 0.189 Normalized Root Mean Square Error (NRMSE). On the other hand, model_finetune(FD_FL) has an error of 0.154, which is between the others.

$$NRMSE = \sqrt{\frac{\Sigma_{i=1}^{n}(l_i - \hat{l}_i)^2}{n(l_1 - l_0)^2}} \tag{4}$$

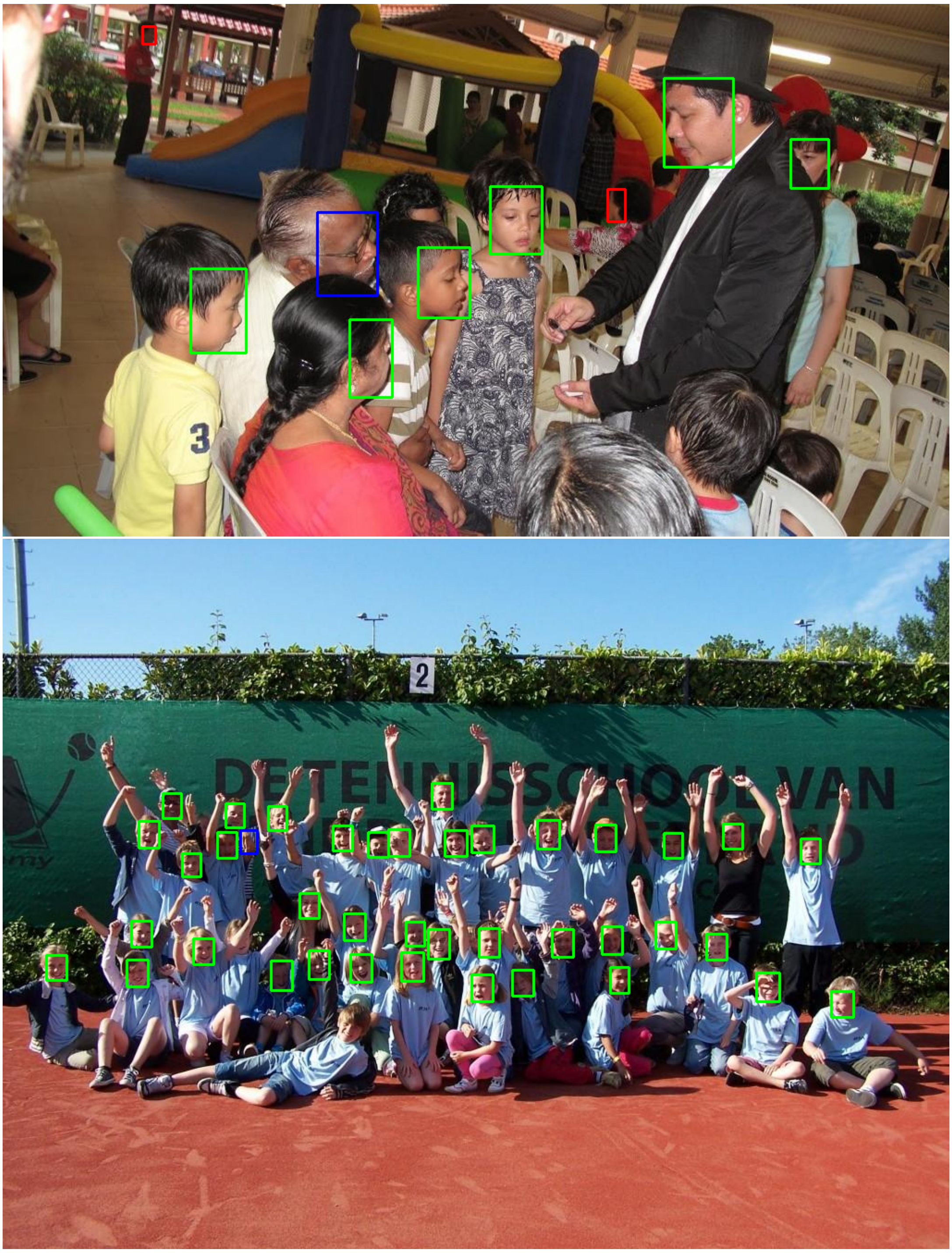


Figure 11: Qulitative results of model_base and model_joint were given on sample images from the Widerface [1] validation set. If the IOU of both models' predicted bounding boxes is greater than 0.5, bounding boxes are shown in green color. Blue rectangles indicate the predictions made only by the model_joint, and the bounding boxes predicted only by the model_base are in red.

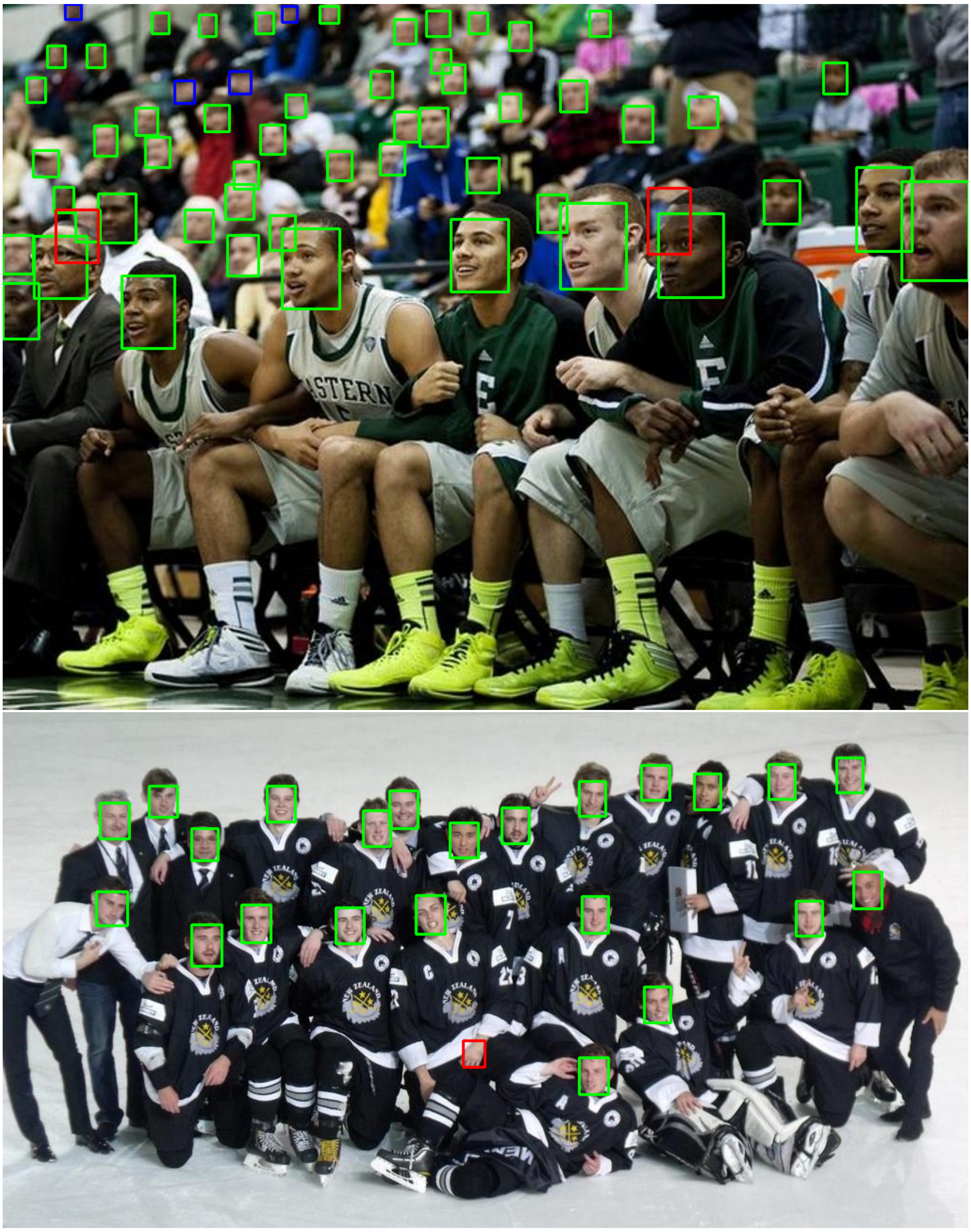


Figure 12: Qulitative results of model_freeze and model_joint were given on sample images from the Widerface [1] validation set. If the IOU of both models' predicted bounding boxes is greater than 0.5, bounding boxes are shown in green color. Blue rectangles indicate the predictions made only by the model_joint, and the bounding boxes predicted only by the model_freeze are in red.

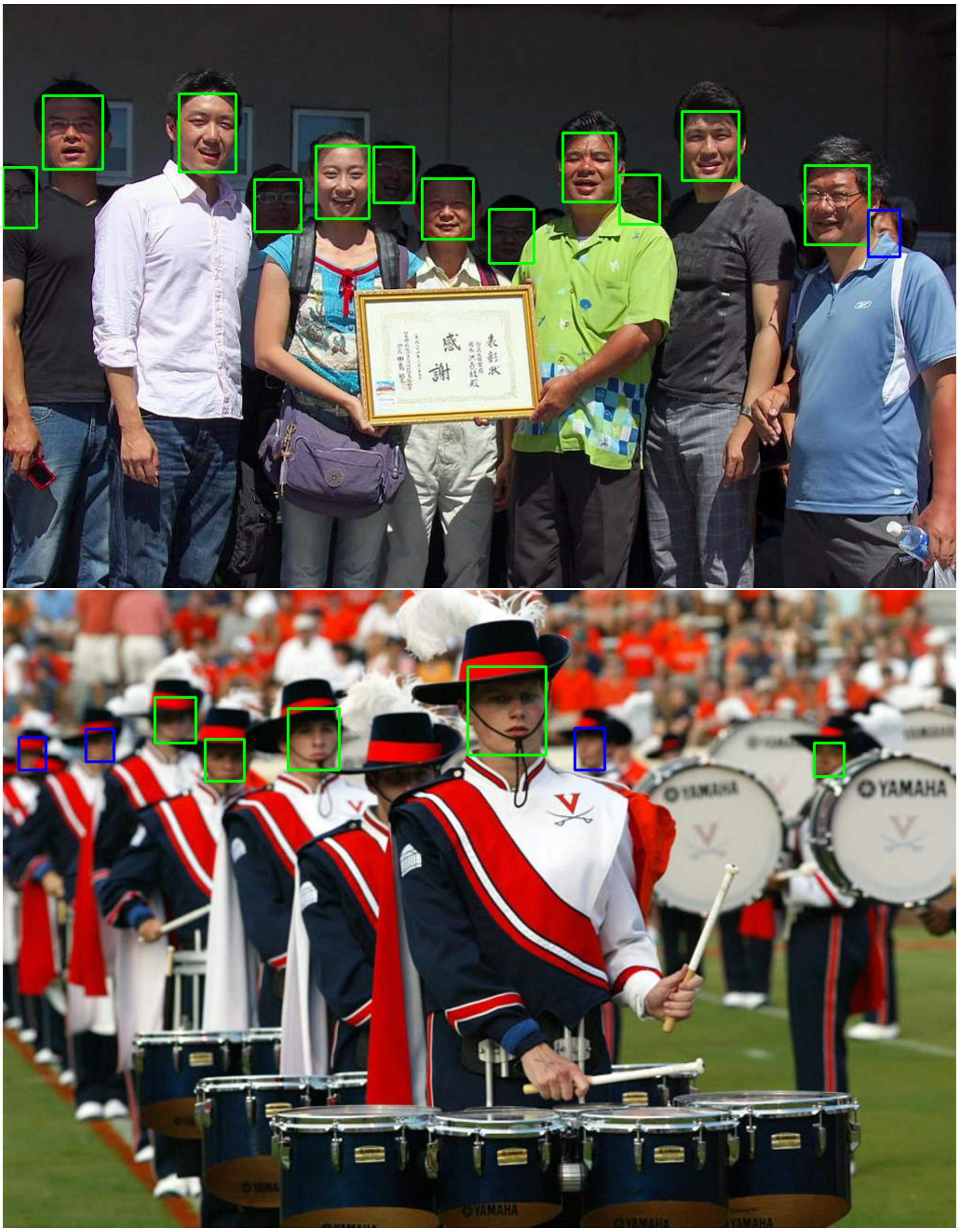


Figure 13: Qulitative results of model_finetune and model_joint were given on sample images from the Widerface [1] validation set. If the IOU of both models' predicted bounding boxes is greater than 0.5, bounding boxes are shown in green color. Blue rectangles indicate the predictions made only by the model_joint, and the bounding boxes predicted only by the model_finetune are in red.

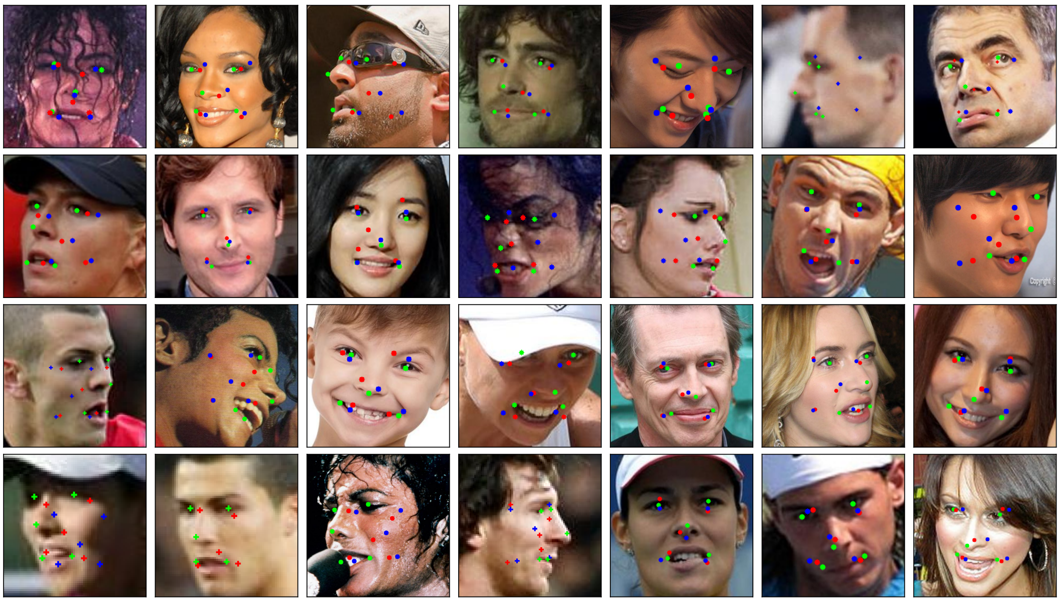


Figure 14: Visualization of predicted facial landmarks on the Widerface [1] validation dataset. Faces were detected by our joint model and were cropped from images. Colors green, blue, and red indicate the predictions of model_joint, model_freeze, and model_finetune, respectively.

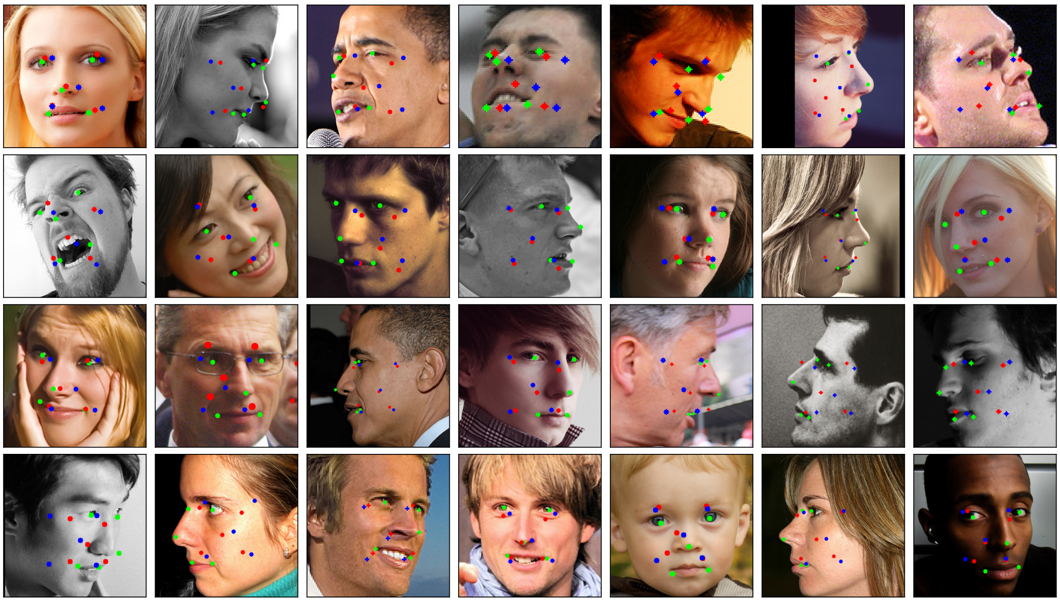


Figure 15: Visualization of predicted facial landmarks on the AFLW [2] test dataset. Faces were detected by our joint model and were cropped from images. Colors green, blue, and red indicate the predictions of model_joint, model_freeze, and model_finetune, respectively.

We also presented qualitative results of compared models in Figure 14 and Figure 15. We visualized landmark predictions of the models on the Widerface validation and the AFLW test sets. All faces were detected by our joint model and then cropped square. As the predicted facial landmarks in Figures14 15 are analyzed, the joint model predicted five facial landmarks significantly better than two other multitask models, model_freeze and model_finetune. Their predictions became worse as faces were profile or heads tilted to the right or left.

3.3 Discussion

In this chapter, the effects of multitask learning on two related tasks (face detection and facial landmark detection) were examined. Firstly, the proposed connectionist model for FD and FL tasks is proposed. The YOLOv5 [5] model was modified for predicting landmarks along with bounding boxes and confidence scores. Then, experimental setups and utilized datasets were presented. Finally, the performance results of the models on two tasks were presented.

The main aim of the experiments was to examine the effect of multitask learning on models' performances for FD and FL tasks. According to experimental results, joint optimization of tasks improved the performances of both tasks, which produced the highest AP in FD task and the lowest NRMSE in FL task. Furthermore, freezing the parameters belonging to old tasks in the new model preserved the performance of old tasks, but the new task could not be learned perfectly. On the contrary, finetuning the old parameters with new tasks degraded the performance of old tasks with the help of better learning of new tasks.

In Chapter 4, different continual learning methods are compared on face analysis tasks. Our MTL model was utilized as an initial step in order to prepare (detecting, aligning, and cropping) faces for face analysis tasks. Thus, we did not focus on the performance of face detection on the hard subset in our experiments.

In conclusion, two similar tasks, which are both regression problems, were jointly learned, and the highest performance results were obtained from the joint model in our experiments.

CHAPTER 4

CONTINUAL LEARNING FOR FACE ANALYSIS

In this chapter, the details of the experiments for the application of different CL methods on face analysis were presented in order to mitigate catastrophic forgetting. While humans can adapt to changing environments and are able to learn sequential experiences, artificial systems are not good at learning from non-stationary data. Although complete forgetting is uncommon in humans, connectionist networks suffer from catastrophic forgetting. As indicated before, face-related tasks, which are age, gender, emotion estimation, and face recognition, are leveraged in this chapter to explore the different CL methods. Firstly, we described the utilized CL approaches in our experiments. Afterward, the experimental setup, utilized datasets, and results of experiments were presented, respectively. Finally, experimental results were discussed in the discussion section.

4.1 Utilized Methods

In the experiments, regularization and generative replay-based methods were employed in order to overcome forgetting. All methods addressed the problem of catastrophic forgetting in neural networks, which occurs when a network is trained on a new task and forgets how to perform a previously learned task. Moreover, fine-tuning and joint, which are usually accepted baselines in CL literature, were also included in comparisons.

All methods were compared on the same model (aka the solver model or the main model), which is indicated in Figure 16. As in [4], the ANN model was designed by taking inspiration from a VAE model since it was utilized for generating and replaying new samples in ANNs like in the human brain. The model consists of 5 or 7 convolutional layers, 4 or 6 of which are followed by Batch Normalization and ReLU activation layers, respectively. In addition, the architecture also includes 3 Fully Connected layers, and the number of output nodes is variable in the final layer. The output prediction size of the model depends on the total number of classes for all tasks in both scenarios. However, all outputs until the current classes are active in the class incremental learning scenario, whereas only the outputs for the classes belonging to the current task are only taken into account in the task incremental learning scenario.

Joint It is actually MTL, which is training a model with all available tasks so far. It is also known as offline training. This setting generally determines the upper bound in CL if no method has the ability of forward transfer.

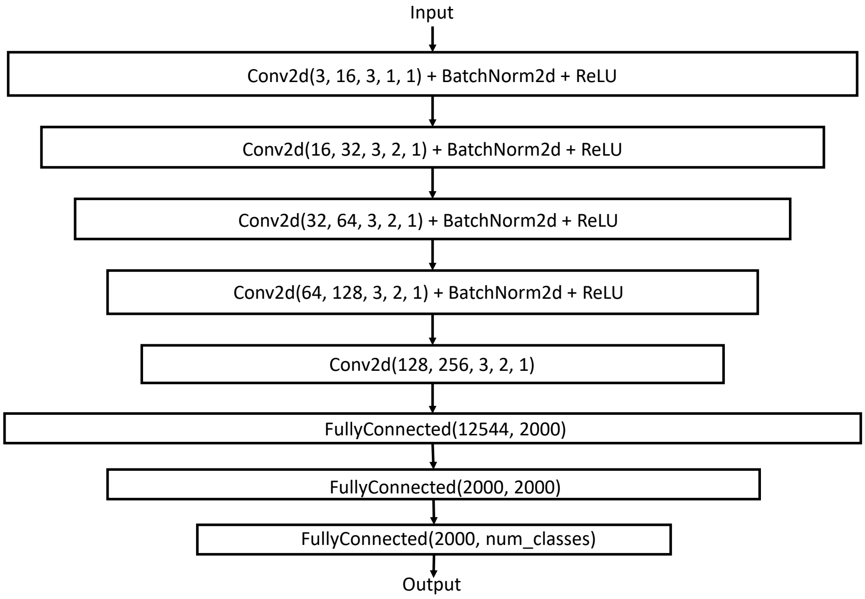

Input
Conv2d(3, 16, 3, 1, 1) + BatchNorm2d + ReLU
Conv2d(16, 32, 3, 2, 1) + BatchNorm2d + ReLU
Conv2d(32, 64, 3, 2, 1) + BatchNorm2d + ReLU
Conv2d(64, 128, 3, 2, 1) + BatchNorm2d + ReLU
Conv2d(128, 256, 3, 2, 1)
FullyConnected(12544, 2000)
FullyConnected(2000, 2000)
FullyConnected(2000, num_classes)
Output

Figure 16: The design of ANN model architecture utilized for face analysis in the experiments.

Fine-tuning (None) It is standard supervised learning, which trains a network by finetuning for a new task and possibly results in catastrophic forgetting. Fine-tuning generally improves the performance of the model for the current task while degrading the performance of the model for the old tasks.

Learning without Forgetting Despite the methods employing weights and change in weights, LwF [81] employed knowledge distillation, which means that a smaller network (called the "student network") is trained to mimic the behavior of a larger network (called the "teacher network"), in order to alleviate forgetting in artificial neural networks. LwF stores predictions of current tasks from the pre-trained model, which is trained with data from previous tasks. Later, it learns new tasks from images and labels of the current task and maintains the performance of old tasks by distilling responses of the previous model for the images of the current task. As in MTL, LwF [81] optimizes parameters for both current and previous tasks using current task data.

Elastic Weight Consolidation EWC [84] addresses this problem by using a quadratic penalty on the weights that are important for old tasks to prevent them from changing too much during training on a new task. EWC adds an additional loss which is weighted by λ to the loss of the current task as defined in Equation 5.

$$L_{total} = L_{current\_task} + \lambda * L_{regularization} \tag{5}$$

Synaptic Intelligence SI [85] improves upon EWC [84] by introducing an additional term to the penalty that takes into account the change in the weights over time rather than just the current values of the weights. In order to accomplish it, they assign an importance factor to each synapse (parameter) by taking into account the importance of each parameter for reducing loss. This allows SI to preserve better the knowledge encoded in the weights and prevent catastrophic forgetting.

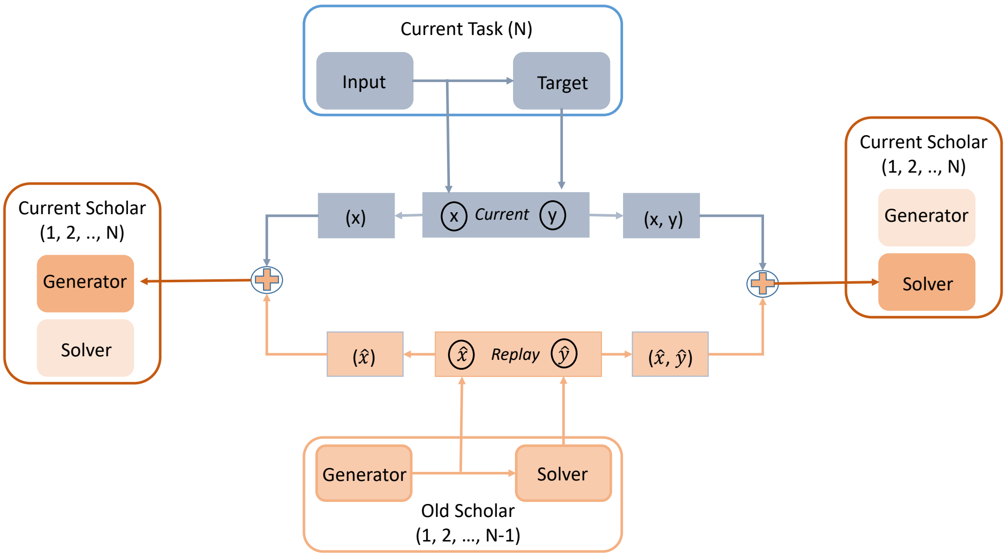


Figure 17: Training of both current generator and current solver for the current task by utilizing old scholar.

In addition to regularization methods, we used methods utilizing generative replay, which generates new samples instead of storing raw samples in our experiments. It is loosely associated with mental images in the human brain.

Deep Generative Replay Generative models refer to models that produce (generate) samples from a distribution. Generative Adversarial Networks and Variational Autoencoders are two well-known methods of generative models. In DGR [112], generative models are utilized for sampling examples from old tasks in order to alleviate catastrophic interference. In the experiments, a symmetric VAE similar to the solver model is chosen as a generative replay model as in [4]. The VAE model maps input images to the latent variables by encoding, and then the latent variables are decoded in order to reconstruct samples. Shin et al. [112] called generator and solver together as a scholar and trained both of them for new tasks(classes), as indicated in Figure 17.

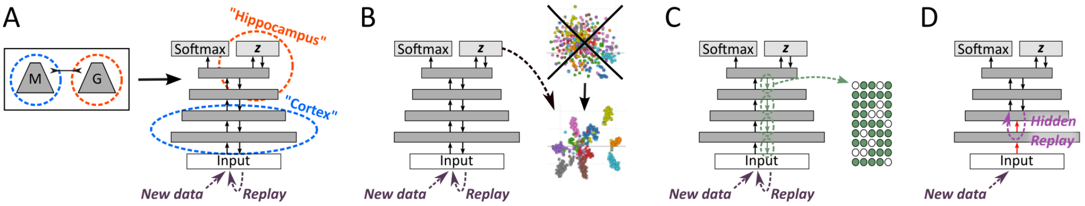


Figure 18: Modifications to the main solver model for the proposed brain-inspired design of [4]. (A) The generator and the solver are coupled with generative feedback connections. (B) A specific class is generated by a Gaussian mixture instead of the normal prior. (C) Different neurons are active during the generative backward pass when classes are learned. (D) The replay occurs through hidden layers to output. The image is taken from [4].

Brain-inspired Replay (BIR) In BIR [4], the authors combined generative and task-solver models into one model by taking inspiration from the human brain. They enriched their design with the addition of generative feedback, internal context gating, and conditional and internal replay, which replays compressed and encoded features instead of raw pixels. Modifications proposed by van de Ven et al. [4] to the main model for BIR are demonstrated in Figure 18. Furthermore, conditional replay facilitates the generation of samples belonging to specific tasks in contrast to random samples.

Brain-inspired Replay + Synaptic Intelligence In class incremental learning scenarios, we also utilized the combination of a replay-based method and a regularization-based method.

4.2 Experiments

In experiments, we compared the CL methods described in the previous section on face analysis tasks according to task incremental and class incremental learning scenarios. In the task incremental learning scenario, the model learns different tasks sequentially and produces outputs only for the specific task. That is, only output nodes of the model for the current task are active. On the other hand, the class incremental learning scenario requires all outputs of the model for previous and current tasks to be utilized in prediction. We utilized the same hardware and software which were introduced in Table 1.

4.2.1 Experimental Setup

We designed 4 different experiments in order to compare mentioned CL methods according to task and class incremental learning scenarios.

Experiment 1 The first experiment was conducted on age, emotion, and gender estimation tasks according to task incremental learning scenarios, in which each task was learned sequentially by the model, and only one task was predicted at the test time. In order to explore whether the task order is important or not in task incremental learning scenarios, we changed the task order and trained 6 different models using 6 different combinations of the task order. CelebA [3] dataset was utilized for training and testing the model. In addition, each task has two different classes; including young and old for the age dataset, smiling or not smiling for the emotion dataset, and male and female for the gender dataset. The heights and widths of all inputs were adjusted to 112 pixels. Furthermore, the cross-entropy loss was used as a loss function due to the classification of inputs between two classes in the training step. The training ran through 5000 iterations for each task, and the performance results were measured after the training with each task.

Experiment 2 In the second experiment, VGGFace2 [6] dataset was used for testing and training the models for both task incremental and class incremental learning scenarios. We took the first 1000 classes(identities) from VGGV2 dataset and split it into 10 tasks which have 100 identities. In class incremental learning protocol, models are responsible for predicting between all classes seen so far. For instance, models produce 1000 different scores after training for 10 tasks. On the other hand, only 100 outputs are always produced in task incremental protocol. Similar to the first experiment, input images had sizes of 112x112 pixels, and the loss function was cross-entropy. Like the previous experiment, the models were trained until 2000 iterations.

Experiment 3 In the third experiment, we utilized 1000 classes of VGGFace2 dataset similar to the experiment 2. In contrast to previous experiments, we increased the number of convolutional layers from 5 to 7. By doing so, we wanted to comprehend whether the larger model is able to learn several tasks better than the smaller model or not. The improved model required inputs with larger sizes due to the fact that each convolutional layer halves the width and height of inputs. Therefore, we adjusted input sizes to 128x128 pixels. We compared CL models using the new enhanced model on both task incremental and class incremental learning scenarios. In addition, we did not change the number of iterations.

Experiment 4 In the last experiment, we increased the number of iterations to 10000 since the model has more convolutional layers, which also leads to an increase in the input size, and a more realistic real-life dataset with several tasks might need to be learned longer wrt. the smaller one with 3 tasks and less number of samples.

In all experiments, we benefited from experimental settings used for CIFAR100 [146] tasks and used default algorithm-specific hyper-parameters in [4]. Similarly, we took the same VAE model, which contains 5 convolutional and 3 fully connected layers, and used a task solver model in experiments 1 and 2. In the third experiment, we employed 7 convolutions and kept the number of fully connected layers the same. As in [4], we froze pre-trained convolutional layers and replayed only internal representations. That is, only fully connected layers were trained with inputs from the last convolution layer. However, we trained convolutional layers with 100 identities from VGGFace2, which are not included in the classes used in the second and third experiments, in contrast to [4], which utilized CIFAR10 [146] datasets for training convolutional layers. In addition, we increased the input sizes of models to 112x112 pixels (Experiments 1 and 2) and 128x128 pixels (Experiments 3 and 4) from 32x32 pixels and replaced CIFAR100 dataset with more complex and real-world datasets, which are CelebA [3] and VGGFace2 [6]. In all experiments, the batch size was chosen as 256.

4.2.2 Utilized Datasets

Large-scale CelebFaces Attributes Dataset

CelebA [3] is a large-scale dataset of celebrity faces that includes 202,599 images of 10,177 different celebrities. Each image in the dataset has been annotated with 40 different binary attributes, such as "male", "smiling", "wearing glasses", and "young" alongside coordinates of bounding boxes and five facial landmarks. The dataset is widely used for research in the field of computer vision and machine learning, particularly for tasks related to facial recognition, face synthesis and attribute prediction. The images in the dataset are all high-resolution and taken from a variety of sources, including movies, television shows, and websites. The celebrities included in the dataset come from a variety of countries and cultures.

Before the CelebA [3] dataset was employed in training for the CL model, we applied some preprocessing to them. First of all, we passed all images through our multitask YOLOv5 face detector and obtained bounding boxes and five landmark points. Subsequently, predicted bounding boxes were compared with ground truth boxes, and the box which has the maximum intersection over union with the ground truth box was selected for aligning and resizing the face to 112x112 pixels. Eventually, we distributed the same number of faces, which is 22,223, to each class of tasks. In total, we have 133,338

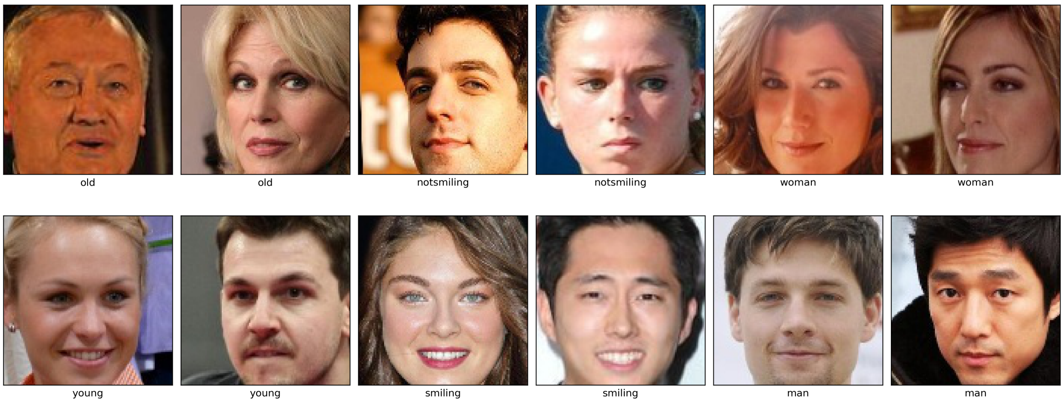


Figure 19: Sample images from CelebA [3] dataset

Figure 20: Sample images from VGGFace2 [6] dataset

faces for three tasks (emotion, age, and gender) and 6 classes (smiling/not smiling, young/old, male/female) (Figure 19). In both the training and testing of models for three tasks, we used CelebA dataset.

VGGFace2 Dataset

VGGFace2 dataset is a large-scale face dataset that includes 3.31 million images from 9131 different subjects. The images were collected from Google search engine and has large variations in pose, age and ethnicity and illumination. The dataset is widely used in face recognition. Similar to CelebA dataset, we used our multitask YOLOv5 model in order to detect faces with landmarks and align faces. After alignment, faces were cropped in 112x112 pixels.

Later, the first 1000 identities were selected, and 10% of them were taken apart for testing while others were used in training. Similarly, additional 100 identities were selected and divided into training and test splits for pretraining convolutional layers of models. Sample images from the dataset are shown in Figure 20.

4.2.3 Experimental Results

In this section, experimental results were presented. To compare different CL methods, different evaluation metrics are available. Lopez-Paz and Ranzato [107] compared their proposed GEM with different CL methods on Average Accuracy (ACC) over all tasks, forward transfer, which measures how the previous task affects the performance of the current task, and Backward Transfer (BWT), which evaluates how the current task influences the performance of the previous task. In addition to performance metrics, measurement of resource consumptions such as utilized or allocated disk space, CPU/GPU, and memory, along with execution time.

$$ACC = \frac{1}{T}\sum_{i=1}^{T} R_{T,i} \tag{6}$$

$$BWT = \frac{1}{T-1}\sum_{i=1}^{T-1} R_{T,i} - R_{i,i} \tag{7}$$

Similar to GEM [107], we utilized the ACC (Equation 6) and BWT (Equation 7) for performance evaluation in our experiments. In order to evaluate performances, we had access to test sets for all tasks T, which helped us to compute the test classification accuracy $R_{i,j}$ measured for task t_j after the model was trained on task t_i. Additionally, we reported the final accuracies $R_{T,i}$, where $i \leq T$ after being trained on all tasks T and plotted the accuracy of each task t_j during training (after each task t_i), where $j \leq i$ and $i, j \leq T$ in order to compare how the methods are successful for mitigating catastrophic forgetting. Positive BWT means that learning new tasks improve the performance of old tasks, while negative BWT occurs when the model forgets old tasks.

In the first experiment, different CL methods were compared using ACC (%) and BWT (%) for task incremental protocol. The model, which included 5 convolutional layers, was trained on Age, Emotion and Gender estimation tasks from CelebA dataset in different sequences. Table 4 indicates average and task-specific ACC (%) after the models were trained for all tasks. Tasks were supplied to the models in different orders and the performance results of each CL method were obtained. According to the results presented in Table 4, all CL methods, including Finetune and Joint performed successfully. In the experiment, CL methods were compared on only the task incremental learning scenario. Since we wanted the model to estimate age, emotion and gender from an image simultaneously, the class incremental learning scenario was not considered in this experiment. In all different task order scenarios, LwF method outperformed all other methods according to average ACC (%). Finetune method exhibited the worst performance in all scenarios. If fine-tuning, which is a baseline method, was not taken into account, EWC was not able to perform well like other methods in EAG EGA GAE GEA task orders. Similarly, when tasks were learned in AEG and AGE orders, GR performed worse slightly wrt. others. In general, BIR enabled the model to produce higher ACC (%) than GR in all scenarios except GEA scenario.

Additionally, we also compared methods by measuring BWT (%) after training was completed on all tasks, as shown in Table 5. When BWT (%) results were examined, all CL methods generally caused negative BWT, meaning that learning new tasks leads to forgetting some old knowledge obtained from old tasks. The largest negative BWTs (%) were obtained when fine-tuning was applied to the models in all sequences of tasks. On the other hand, the models learned new tasks with the highest BWT (%) in all scenarios when LwF method was utilized during training. Additionally, LwF exhibited positive

Table 4: Comparison of different CL methods using ACC (%) according to task incremental learning scenario on Age (A), Emotion (E) and Gender (G) estimation tasks from CelebA [3] dataset after the model with 5 convolutional layers is trained with each task in different orders of tasks. The models were trained in 5000 iterations for each task.

Method	ACC			Task Order																	
				AEG			AGE			EAG			EGA			GAE			GEA		
Joint	**Average**			89.08			89.56			89.34			89.34			89.78			89.45		
	A	**E**	**G**	80.84	90.19	96.20	81.39	90.42	96.85	80.66	91.05	96.31	80.45	91.09	96.47	81.65	90.94	96.76	80.70	90.71	96.94
Finetune	**Average**			84.74			81.81			82.98			82.82			82.99			83.93		
	A	**E**	**G**	72.06	84.95	97.19	67.81	91.84	85.78	74.81	76.90	97.23	82.59	74.40	91.45	76.02	91.66	81.29	82.59	82.64	86.55
LwF	**Average**			**89.98**			**89.73**			**89.92**			**90.05**			**90.31**			**90.50**		
	A	**E**	**G**	81.62	91.54	96.76	81.17	90.80	97.21	81.35	91.70	96.69	81.31	91.70	97.14	82.34	91.23	97.37	82.42	91.61	97.48
SI	**Average**			88.45			89.08			88.68			88.54			87.37			88.69		
	A	**E**	**G**	80.16	89.07	96.11	80.93	89.59	96.72	80.25	90.22	95.57	79.35	90.37	95.88	77.60	88.78	95.75	80.59	89.38	96.09
EWC	**Average**			89.11			89.00			87.69			88.06			83.69			88.27		
	A	**E**	**G**	80.41	90.55	96.38	80.41	90.89	95.70	76.65	90.06	96.36	80.54	88.73	94.92	74.85	88.69	87.52	80.86	88.44	95.52
GR	**Average**			85.89			86.77			87.99			88.48			88.29			89.14		
	A	**E**	**G**	71.68	89.59	96.40	74.94	89.34	96.02	79.22	89.05	95.70	80.21	89.00	96.22	79.19	89.86	95.82	80.84	90.64	95.95
BIR	**Average**			88.97			88.78			88.74			89.07			88.67			88.82		
	A	**E**	**G**	79.58	90.69	96.65	79.82	90.04	96.47	79.49	90.55	96.18	80.18	90.62	96.40	79.98	89.88	96.15	79.55	90.71	96.20

Table 5: Comparison of different CL methods using BWT (%) according to task incremental learning scenario on Age (A), Emotion (E) and Gender (G) estimation tasks from CelebA [3] dataset after the model with 5 convolutional layers is trained with each task in different orders of tasks. The models were trained in 5000 iterations for each task.

Method	Task Order					
	AEG	AGE	EAG	EGA	GAE	GEA
Joint	-0.21	0.48	-0.25	0.01	-0.18	-0.28
Finetune	-7.74	-12.26	-10.50	-10.84	-11.16	-9.38
LwF	**0.73**	**0.92**	**1.16**	**0.55**	**-0.10**	**0.35**
SI	-0.28	0.18	-0.22	-0.04	-1.28	-0.55
EWC	-0.31	-0.84	-2.55	-1.81	-7.80	-1.81
GR	-5.07	-3.63	-0.87	-0.90	-1.38	-0.93
BIR	-0.25	-0.24	-0.17	0.02	-0.82	-0.21

backward transfer in all task sequences except GAE order. That is, learning new tasks improved the performance of the old tasks, which is uncommon.

In the second experiment, we compared various continual learning methods using average accuracy and backward transfer in task incremental and class incremental learning scenarios for 10 tasks on the VGGFace2 dataset. The model had 5 convolutional layers. Firstly, we calculated the average accuracy on all tasks seen so far in order to measure how well the model learns new tasks without forgetting old ones. As shown in Figure 21, all methods except GR and Finetune could learn tasks successfully in the task incremental learning scenario. Furthermore, the Joint model had the highest accuracy mostly in both scenarios. According to the results of the class incremental learning scenario, the average accuracy over tasks decreased gradually as the number of classes to be learned increased. Albeit low performances of BIR and SI separately, utilization of SI with BIR enhanced ACC (%), and it had the highest average accuracy as a CL method. Furthermore, ACC and BWT were calculated after the model was trained on all tasks to compare methods on final average accuracy and backward transfer after all tasks were learned. In the task incremental learning scenario, 100 out of 1000 output nodes of the model were active, and they were selected wrt. the current task. The ACC (%) performance of the model is upper bounded with the joint method, whereas the lower bound is obtained from fine-tuning

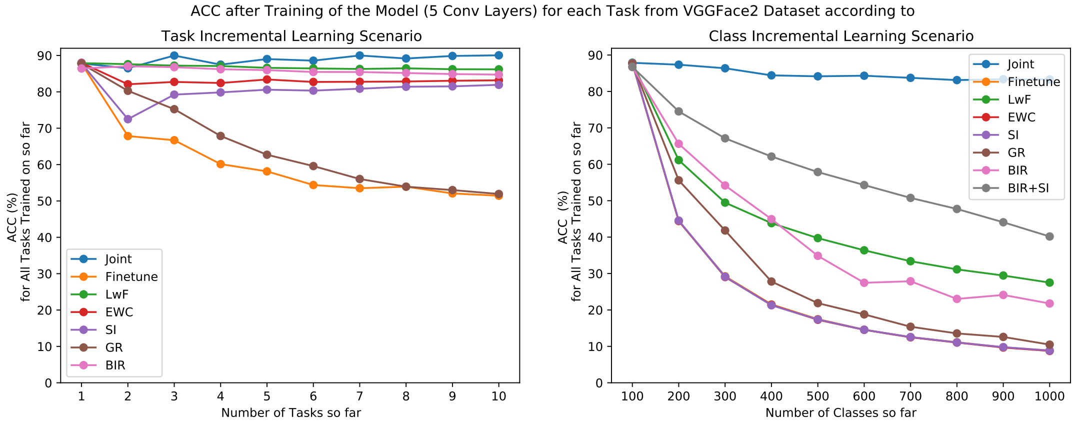


Figure 21: Comparison of different CL methods using ACC (%) for task incremental learning scenario (**Left**) and class incremental learning scenario (**Right**) for 10 tasks/episodes on VGGFace2 [6] dataset after training of the model with 5 convolutional layers is completed on all tasks. The models were trained in 2000 iterations for each task.

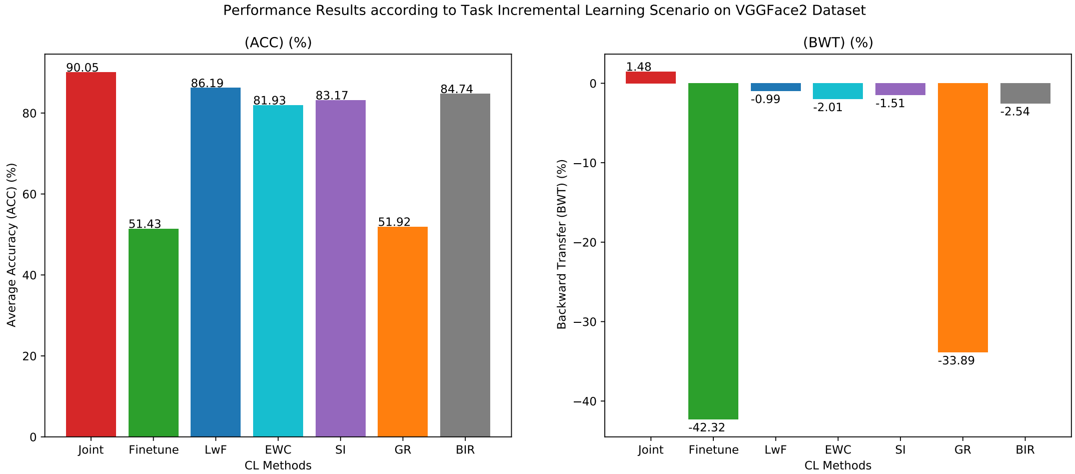


Figure 22: Comparison of different CL methods using ACC (%) (**Left**) and BWT (%) (**Right**) according to task incremental learning scenario for 10 tasks/episodes on VGGFace2 [6] dataset after training of the model with 5 convolutional layers is completed on all tasks. The models were trained in 2000 iterations for each task.

(Figure 22 **Left**). Other methods apart from GR performed similarly. When BWT (%) in Figure 22 (**Right**) is compared with the results of the first experiment, Finetune and GR forgot old information ensuing old tasks significantly as new tasks were added to the model. However, Figure 22 (**Right**) shows dramatic negative backward transfer in both Finetune and GR methods compared with the first experiment. On the other hand, BIR, which is a generative replay method, was found to be more effective at preventing catastrophic forgetting in the model compared to GR, and the joint method had the best performance with the positive backward transfer.

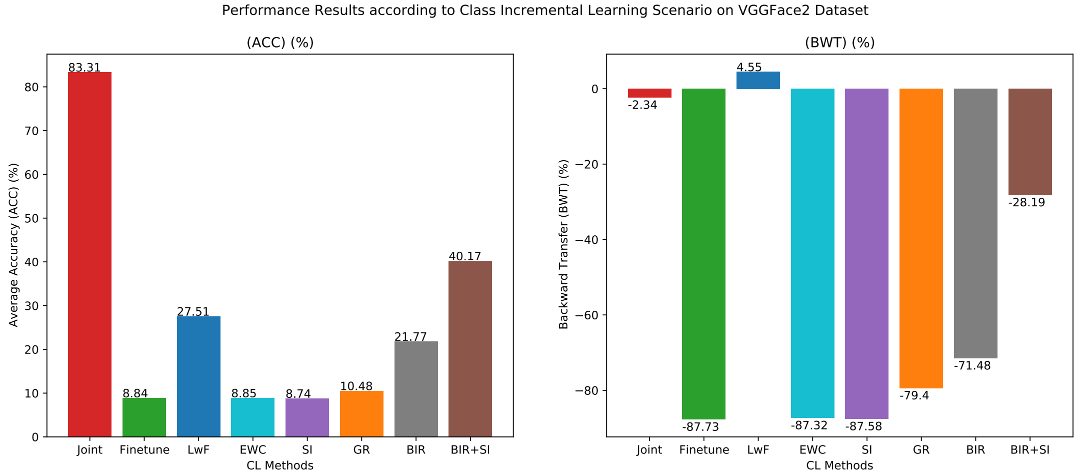


Figure 23: Comparison of different CL methods using ACC (%) (**Left**) and BWT (%) (**Right**) according to class incremental learning scenario for 10 tasks/episodes on VGGFace2 [6] dataset after training of the model with 5 convolutional layers is completed on all tasks. The models were trained in 2000 iterations for each task.

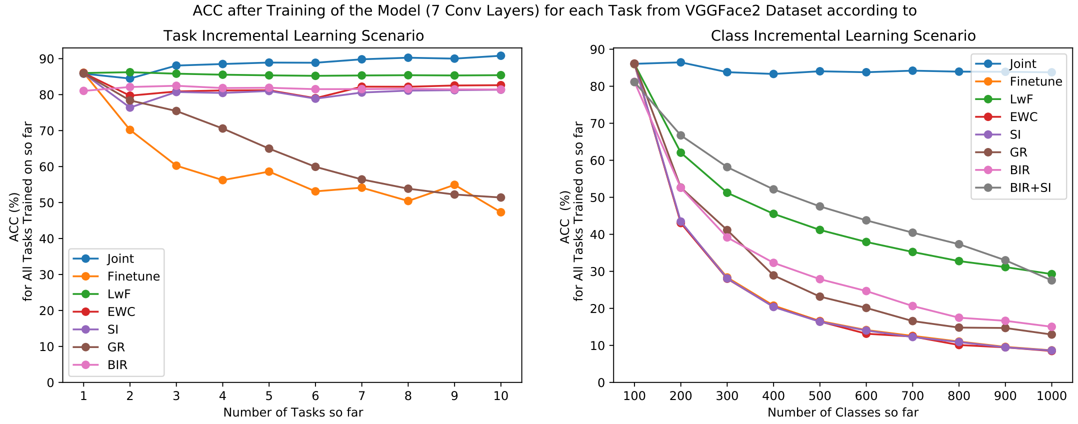


Figure 24: Comparison of different CL methods using ACC (%) according to task incremental learning scenario (**Left**) and class incremental learning scenario (**Right**) for 10 tasks/episodes on VGGFace2 [6] dataset after training of the model with 7 convolutional layers is completed on all tasks. The models were trained in 2000 iterations for each task.

In contrast to the task incremental learning scenario, class incremental protocol resulted in catastrophic forgetting in all CL methods (except the joint method) (Figure 23). Although LwF had positive BWT, it performed poorly. Nevertheless, the combination of BIR and SI methods revealed a better alternative method in order to prevent forgetting in the class incremental learning scenario.

In the third experiment, we increased the number of convolutional layers to 7. The aim of this experiment was whether the model was able to learn tasks better or not if the model had a larger capacity for learning. When Figure 24 (Left) and Figure 21 (Left) are examined, there was no significant difference in the performance of the methods when applied to the small and large models in task incremental learning scenarios. On the other hand, the performances of BIR and BIR+SI degraded in contrast to others in class incremental learning scenarios as additional capacity was added to the models (Figures 24 (Right) and 21 (Right)). Additionally, it is observed from Figure 25 that all CL methods demonstrated inferior performance as compared with the joint method in the task incremental learning scenario. Although BWT (%) for LwF, EWC, SI, GR and BIR increased, meaning forgetting was reduced while learning new tasks, ACC (%) did not improve. In other words, the models could not learn adequately despite their extra available capacity. This suggests that the number of iterations was insufficient in order for the models to acquire new knowledge from the current task. On the other hand, the results presented in Figure 26 show that joint and LwF methods benefitted from extra layers in the model, whereas others were negatively affected in the class incremental learning scenario. In contrast to BIR and BIR+SI methods, other continual learning methods reduced the negative effect of learning new tasks on previously acquired knowledge, which is interpreted from the BWT (%) introduced in Figure 26.

In the last experiment, the increment in the number of iterations improved the efficiency of the joint method slightly. On the contrary, Figure 27 (Left) indicates that the performances of the models regressed dramatically as the new tasks were learned by fine-tuning. Moreover, other methods were also slightly affected negatively in the task incremental learning scenario. In addition, longer training iterations enabled the GR method to improve the performance of the model substantially, as shown in Figure 27 (Right). As we examined the results indicated in Figure 28, ACC (%) got better when only the joint method was employed, and all other methods resulted in lower average accuracy when the number of iterations was made 10000 in task incremental learning scenario. Similarly, BWT (%) also deteriorated in all methods apart from GR. On the other hand, methods except Finetune and LwF took advantage of longer iterations during training for class incremental learning scenarios as observed in Figure 29. As training iterations were changed to 10000, all methods resulted in lower BWT (%). In other words, the negative effects of learning new tasks on the models declined.

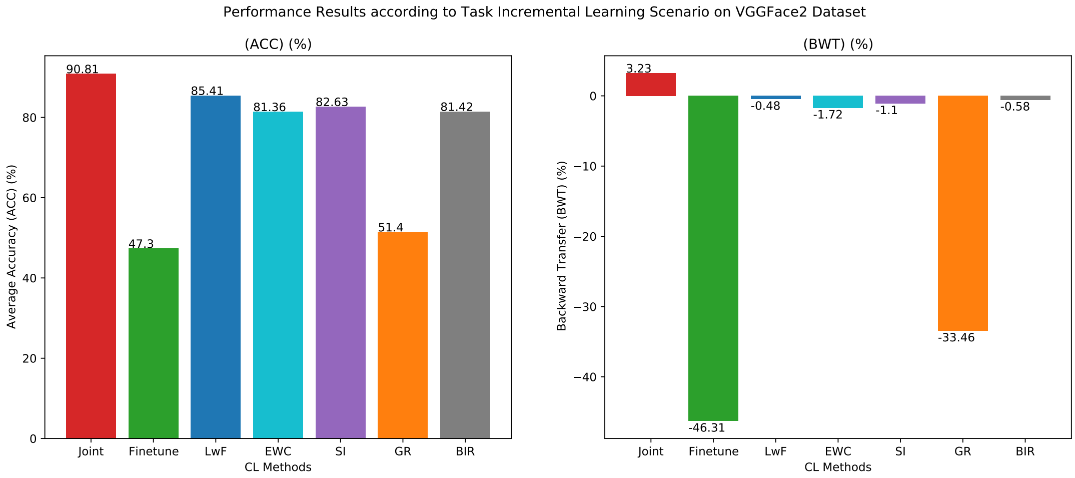


Figure 25: Comparison of different CL methods using ACC (%) (**Left**) and BWT (%) (**Right**) according to task incremental learning scenario for 10 tasks/episodes on VGGFace2 [6] dataset after training of the model with 7 convolutional layers is completed on all tasks. The models were trained in 2000 iterations for each task.

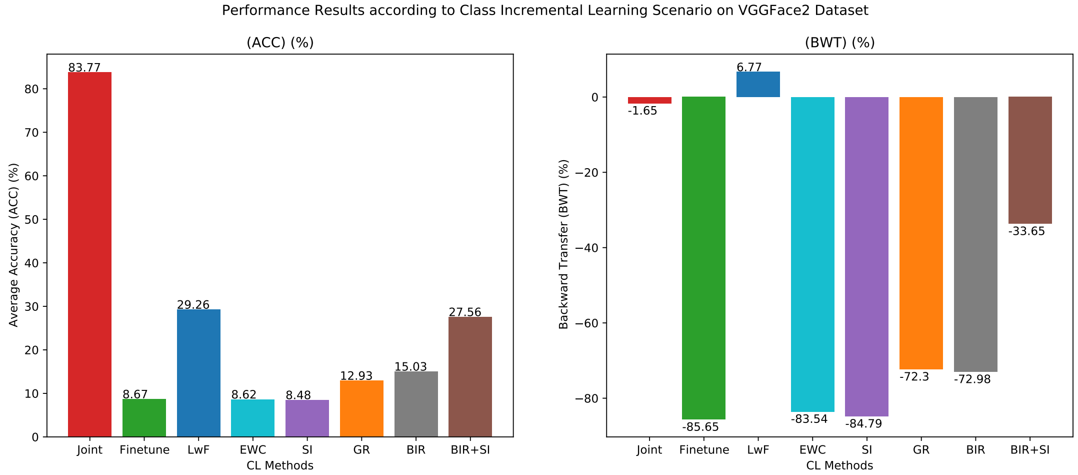


Figure 26: Comparison of different CL methods using ACC (%) (**Left**) and BWT (%) (**Right**) for the class incremental learning scenario for 10 tasks/episodes on VGGFace2 [6] dataset after training of the model with 7 convolutional layers is completed on all tasks. The models were trained in 2000 iterations for each task.

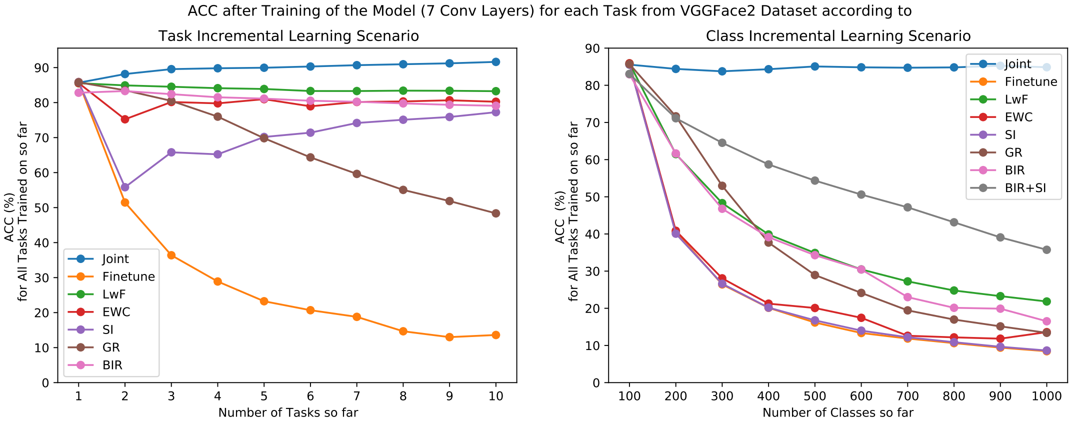


Figure 27: Comparison of different CL methods using ACC (%) according to task incremental learning scenario (**Left**) and class incremental learning scenario (**Right**) for 10 tasks/episodes on VGGFace2 [6] dataset after training of the model with 7 convolutional layers is completed on all tasks. The models were trained in 10000 iterations for each task.

While training the task solver model by employing generative replay, we also trained the generator, which was the VAE model, including 5 or 7 convolutional and 3 fully connected layers. Since generative replay-based methods do not store raw images, images are generated during the training of the task solver model. Figure 30 shows randomly generated sample faces after finishing the training of both generator and task solver models, which were used in the GR method. The sample images were added to the input batches along with images from the current task in order to train the solver model for age, emotion, and gender estimation tasks. Furthermore, Figure 31 presents random face samples generated by the VAE with 5 convolutional layers for face recognition tasks. As shown in the figure, some faces lack facial details, which might degrade the discriminative performance of the

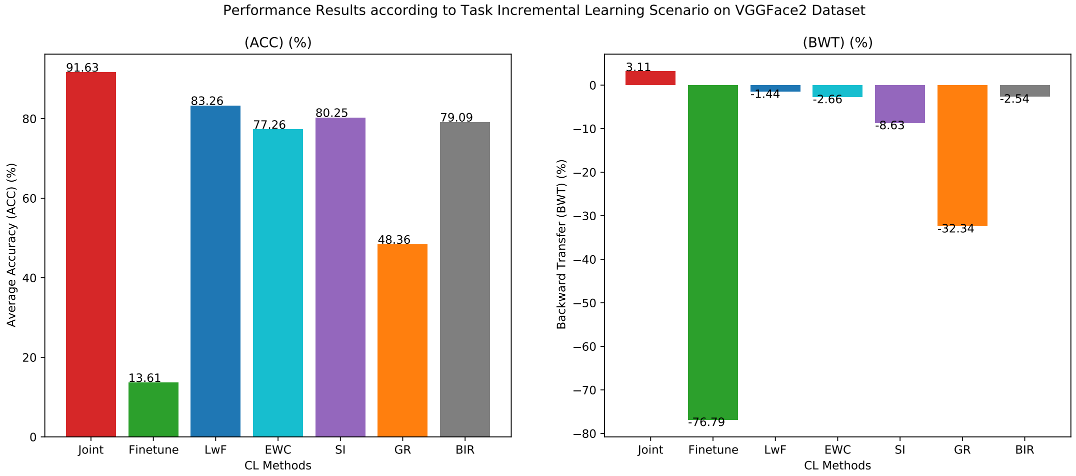


Figure 28: Comparison of different CL methods using ACC (%) (**Left**) and BWT (%) (**Right**) for the task incremental learning scenario for 10 tasks/episodes on VGGFace2 [6] dataset after training of the model with 7 convolutional layers is completed on all tasks. The models were trained in 10000 iterations for each task.

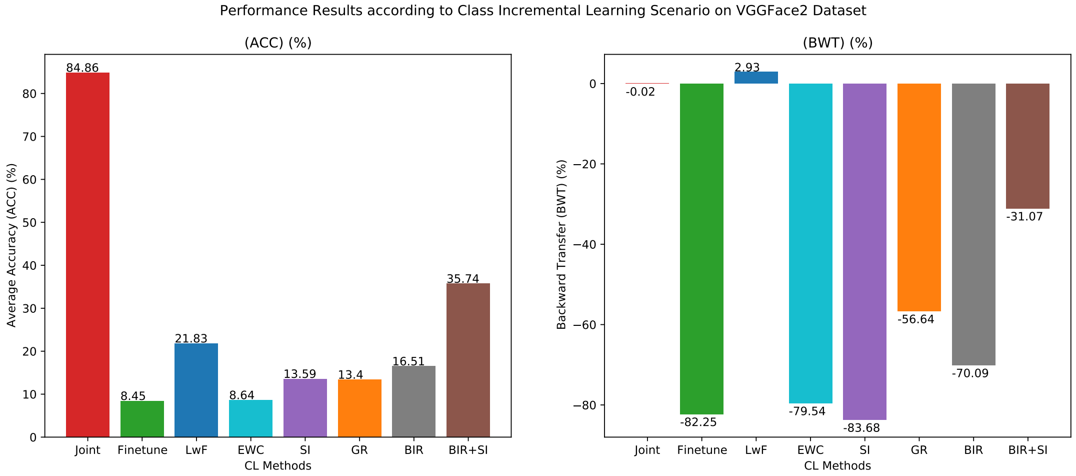


Figure 29: Comparison of different CL methods using ACC (%) (**Left**) and BWT (%) (**Right**) for the class incremental learning scenario for 10 tasks/episodes on VGGFace2 [6] dataset after training of the model with 7 convolutional layers is completed on all tasks. The models were trained in 10000 iterations for each task.

solver model. In contrast, the VAE models with 7 convolutional layers generated more detailed and higher-quality face images, which is presented in Figure 32 and Figure 33 compared to its counterpart with 5 convolutional layers. Upon comparison of the backward transfer of the GR method in the third experiment with that of the second experiment, the former demonstrated a higher percentage of BWT. This is attributed to the higher-quality input data generated during training. Additionally, longer training of the VAE model also enhanced the quality of the generated images in both task incremental and class incremental learning scenarios.

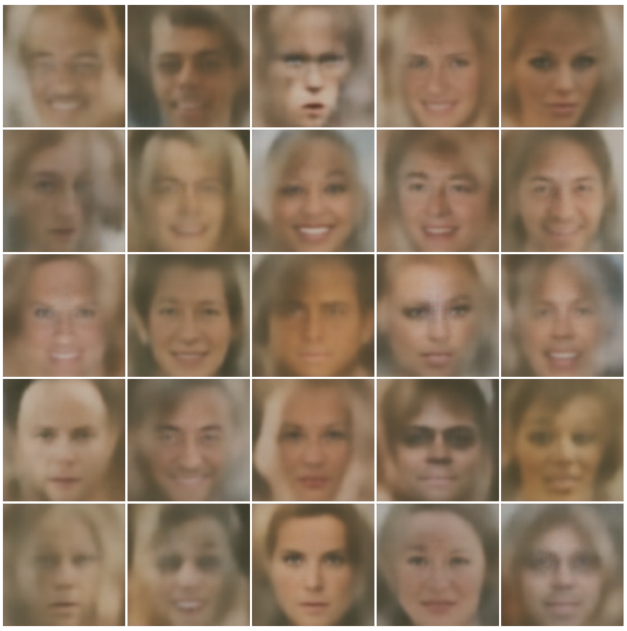
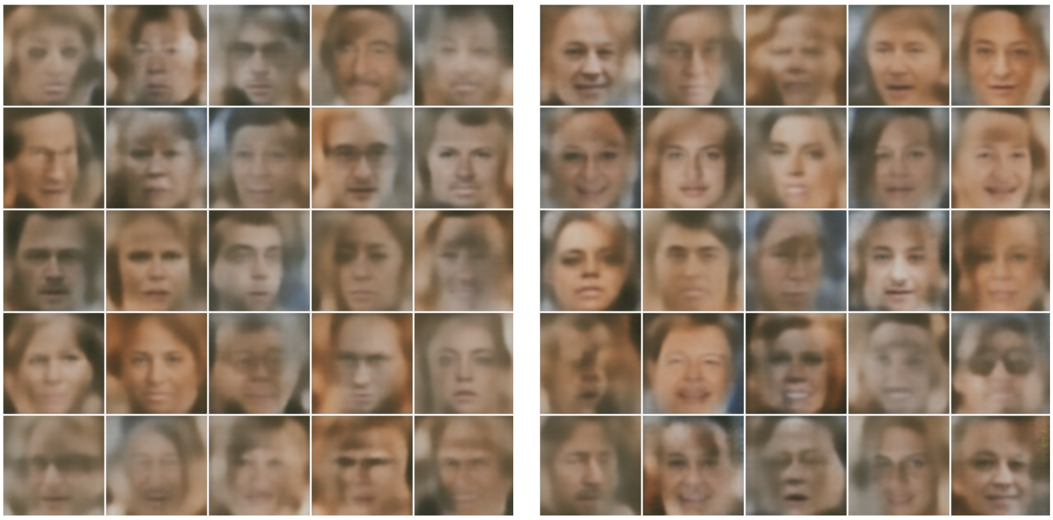


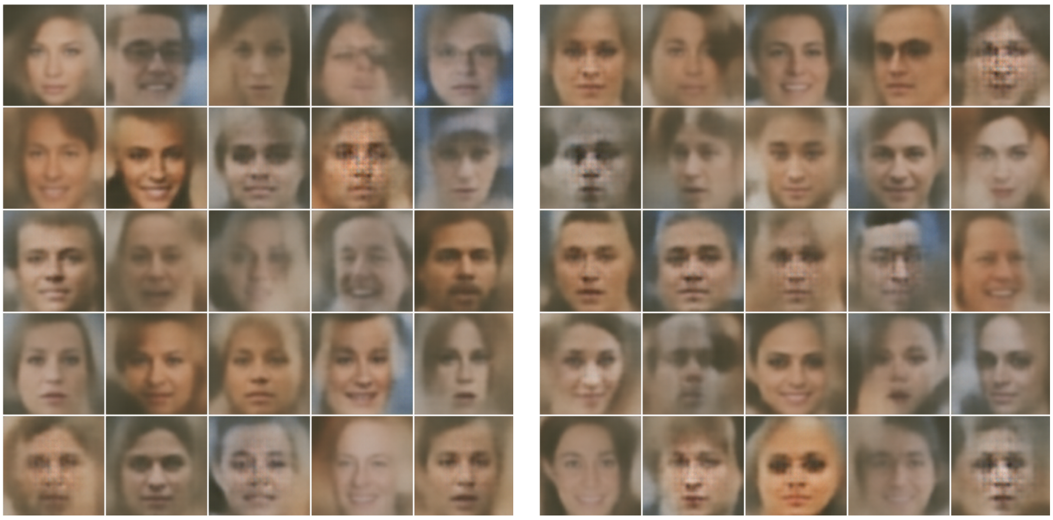
Figure 30: Sample images generated for the training of Age, Emotion and Gender estimation tasks by the VAE with 5 convolutional layers. The VAE model was trained on CelebA dataset according to the task incremental learning scenario in 5000 iterations.



(a) Task Incremental Learning Scenario

(b) Class Incremental Learning Scenario

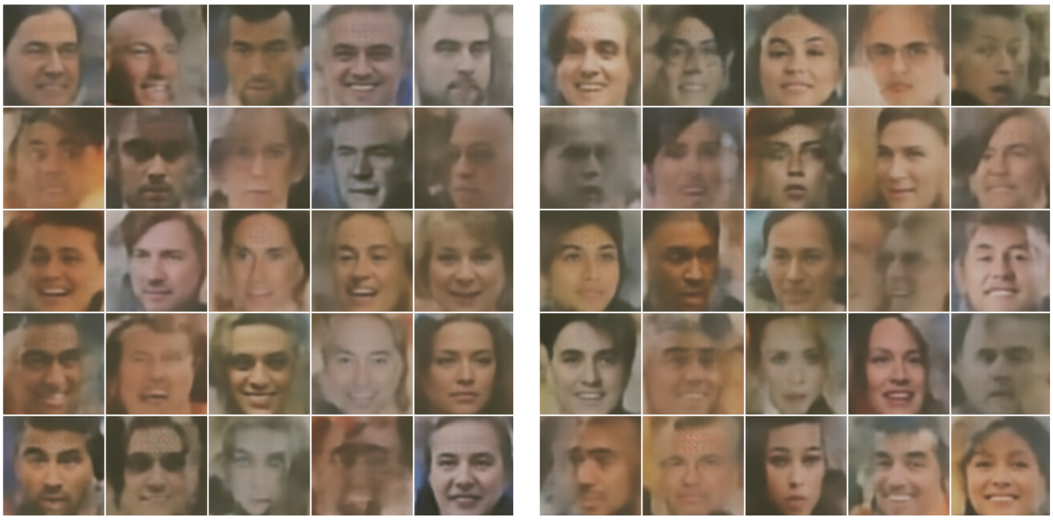
Figure 31: Sample images generated for the training of face recognition tasks by the VAE with 5 convolutional layers. The VAE model was trained on VGGFace2 dataset for task incremental (a) and class incremental (b) learning scenarios in 2000 iterations.



(a) Task Incremental Learning Scenario

(b) Class Incremental Learning Scenario

Figure 32: Sample images generated for the training of face recognition tasks by the VAE with 7 convolutional layers. The VAE model was trained on VGGFace2 dataset for task incremental (a) and class incremental (b) learning scenarios in 2000 iterations.



(a) Task Incremental Learning Scenario

(b) Class Incremental Learning Scenario

Figure 33: Sample images generated for the training of face recognition tasks by the VAE with 7 convolutional layers. The VAE model was trained on VGGFace2 dataset for task incremental (a) and class incremental (b) learning scenarios in 10000 iterations.

4.3 Discussion

In this chapter, different CL methods were analyzed for age estimation, emotion recognition, and gender classification according to task incremental learning scenario on a custom CelebA dataset. Furthermore, we further examined the methods for split face recognition tasks according to both task and class incremental learning protocols on VGGFace2 dataset. Firstly, utilized regularization-based and replay-based CL methods and datasets were presented. Later, experimental results were presented.

The first experimental results indicated that all five CL methods mitigate catastrophic forgetting and learn new tasks successfully in the task incremental learning scenario. Also, LwF [81] outperformed the other four methods on three tasks (age, emotion and gender recognition). According to experimental findings on split face recognition tasks, all methods except Finetune and GR were able to alleviate catastrophic forgetting while learning 10 tasks according to the task incremental learning scenario. Unfortunately, all investigated CL methods could not prevent both VAE models with 5 and 7 convolutional layers from forgetting during the class incremental learning protocol. Nevertheless, the larger VAE model facilitated the generation of high-quality faces, which helped GR methods to perform slightly better in both task and class incremental learning scenarios. Furthermore, the utilization of BIR, along with SI resulted in the remarkable improvement of ACC (%). In addition, increasing the number of training iterations enabled the methods apart from Finetune and LwF to improve the ACC (%) of the models along with higher backward transfer in class incremental learning considerably.

In conclusion, more work is needed in order to alleviate catastrophic forgetting together with increasing backward transfer in class incremental learning scenarios, and a combination of different CL methods might be a solution to accomplish it.

CHAPTER 5

SUMMARY AND CONCLUSIONS

5.1 Summary

In this thesis, two brain-inspired machine learning paradigms, multitask and continual learning approaches, were investigated for face analysis tasks including face detection, landmark extraction, age estimation, emotion recognition, gender classification, and face recognition. In the proposed two-stage framework, face and landmark detection were performed in the first stage. Later, age, emotion, and gender analysis or face recognition were accomplished on detected faces in the second stage.

In Section 2.1, different machine learning paradigms such as supervised learning, transfer learning, domain adaptation, curriculum learning, etc., which take inspiration from the way the human brain processes information, were introduced. Subsequently, previous studies related to multitask learning were given in a general overview briefly in Section 2.2.1. The studies were categorized into two: hard and soft parameter sharing methods. Furthermore, Section 2.2.2 presented the recent studies in continual learning literature within three groups, which are regularization-based, parameter isolation-based, and replay-based methods. Finally, MTL and CL approaches dedicated to face analysis tasks face detection, landmark extraction, gender recognition, and age estimation were introduced in Section 2.2.3.

In Chapter 3, details of the proposed MTL approach that unified detection and landmark extraction tasks, and enabled simultaneous processing were presented. After introducing the datasets and experimental setups for these tasks, we provided in-depth analyses for singletask and multitask models as well as the addition of an auxiliary task (facial landmark extraction) to the initial main face detection task. Additionally, we conducted experiments to train our previously optimized face detection model to also perform facial landmark extraction. This was achieved either by freezing the parameters used for the face detection task or by fine-tuning all parameters for a new task. Finally, we presented and interpreted our experimental results.

In Chapter 4, different CL methods were compared on age estimation, emotion recognition, and gender recognition according to task incremental learning scenarios. Additionally, experiments on task order were conducted by shuffling the tasks to observe the effect of different combinations on performances during the training stage. Custom CelebA dataset was prepared for the experiments by cropping and aligning faces with bounding boxes and landmark points obtained from our MTL model. Later, CelebA dataset was divided into three different tasks, and the performance of the models was measured on them. In order to examine the performances of CL methods, we increased the number of tasks by employing VGGFace2 dataset for split face recognition tasks. Furthermore, both task and class incremental learning scenarios were carried out in the experiments. Finally, experimental findings were presented using average accuracy and backward transfer metrics.

5.2 Conclusions

The problem of face analysis is investigated in two main steps. In the first step, face detection and facial landmark detection tasks are analyzed via multitask settings. Experimental results indicate that the joint MTL model, which is optimized for both face detection and landmark extraction at the same time, outperforms the single-task model on the face detection task. One interpretation is that MTL leveraged the performance of tasks that are similar and related, and have the same type of loss function definitions.

We also utilize the methods of weight freezing and weight fine-tuning when facial landmarks detection is incorporated with face detection. As we examine the experimental results, joint optimization of tasks also outstrips freezing and fine-tuning methods in both detection and landmark detection tasks. In other words, the training of multiple tasks together exposes the intrinsic information contained in each task, and the MTL model benefits from information coming from both tasks. The fine-tuning method outperforms the freezing method on the facial landmark detection task since it has more parameters to be learned. However, the error between predicted and ground truth landmarks is higher compared to the error produced by the jointly optimized model. Although multitask learning is an optimum solution for learning and inferring multiple tasks, obtaining datasets with labels for all tasks is very compelling, and methods for learning new tasks are needed to catch the performance of MTL.

Moreover, we compare different CL methods on CelebA and VGGFace2 datasets in the second stage. The models are trained using age estimation, emotion recognition, and gender classification tasks according to task incremental learning scenarios. When humans learn new concepts, they first learn basic concepts before moving on to more complex topics. As a similar approach, we alter the order of tasks and train 6 different models in order to analyze whether the order of tasks is important in our experiments. However, experimental results indicate that the ACC (%) performances are generally close to each other, and the order of age, emotion, and gender recognition tasks is not important. According to both ACC (%) and BWT (%) performances, LwF method outperforms its counterparts in all task order scenarios. Additionally, LwF method enables the model to improve the performance for previously learned tasks with positive backward transfer in all task orders except the GAE order. The largest forgetting occurs during acquiring new tasks when fine-tuning method is utilized. Unlike fine-tuning, all methods reduce forgetting on CelebA for task incremental scenarios. In addition, BIR performs better than GR in backward transfer along with higher ACC for task incremental scenarios.

Later, CL methods are benchmarked by increasing the number of tasks from 3 to 10, which are obtained by splitting 1000 identities into 10 episodes from VGGFace2 dataset. In task incremental learning scenarios, all methods except fine-tuning and GR mitigate catastrophic forgetting successfully. When the number of convolutional layers in the VAE model is increased from 5 to 7 along with additional 8k training iterations, the joint (MTL) method increases the ACC (%) whereas the performances of all other methods are degraded significantly. The BWT (%) results show that information acquired from old tasks is forgotten dramatically in the fine-tuning method. On the other hand, GR increases the backward transfer of new knowledge faintly.

When methods are compared according to class incremental learning scenarios, all methods apart from the joint method underperform compared to performances in task incremental learning scenarios. Despite the low performances of BIR and SI separately, the utilization of BIR, along with SI results in the remarkable improvement of ACC (%). Moreover, more iterations improve the average accuracy along with higher backward transfer in class incremental learning protocols, excluding Finetune and LwF.

Finally, all investigated CL algorithms reduce forgetting especially in task incremental learning scenarios. LwF, which distills knowledge with soft labels obtained from the current data via the pretrained model to the current model, outperforms all single methods in both scenarios. The joint method, which is actually MTL, is accepted as upper bound performance and is the most suitable solution when attaining new tasks in ANNs. However, data is non-stationary and does not have available labels for all tasks at the same time. Therefore, partial replay-based methods emulate MTL by utilizing some samples from old tasks together with current task data. Similarly, generative replay methods follow a similar approach while solving the privacy concern of storing old samples in partial replay methods. In the experiments, GR usually underperforms particularly in backward transfer, but the larger VAE improves GR methods for both task and class incremental scenarios. That is, the effectiveness of GR can be boosted by utilizing more advanced architecture such as Generative Adversarial Networks or Stable Diffusion models. Although generative replay-based methods are biologically more plausible, regularization-based methods are sufficient for task incremental learning protocols principally.

In conclusion, the class incremental learning scenario requires new studies for mitigating forgetting in connectionist networks while learning tasks from more realistic data. In order to accomplish the performance of task incremental learning scenario in a class incremental learning scenario, combinations of more brain-inspired continual learning methods are needed.

REFERENCES

[1] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5525–5533, 2015.

[2] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 2144–2151, 2011.

[3] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[4] G. M. van de Ven, H. T. Siegelmann, and A. S. Tolias, "Brain-inspired replay for continual learning with artificial neural networks," *Nature Communications*, vol. 11, 2020.

[5] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, K. Michael, TaoXie, J. Fang, imyhxy, Lorna, Z. Yifu, C. Wong, A. V, D. Montes, Z. Wang, C. Fati, J. Nadar, Laughing, UnglvKitDe, V. Sonck, tkianai, yxNONG, P. Skalski, A. Hogan, D. Nair, M. Strobel, and M. Jain, "ultralytics/yolov5: v6.0 - YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support," Oct. 2021.

[6] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 67–74, 2017.

[7] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in Cognitive Sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[8] M. McCloskey and N. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychology of Learning and Motivation - Advances in Research and Theory*, vol. 24, pp. 109–165, Jan. 1989.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.

[10] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[11] X. Guo, S. Singh, H. Lee, R. L. Lewis, and X. Wang, "Deep learning for real-time atari game play using offline monte-carlo tree search planning," in *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds.), vol. 27, Curran Associates, Inc., 2014.

[12] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, Jan. 2016.

[13] M. Mermillod, A. Bugaiska, and P. Bonin, "The stability-plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects," *Frontiers in Psychology*, vol. 4, 2013.

[14] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[15] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep. 0, University of Toronto, Toronto, Ontario, 2009.

[16] T. Allison, A. Puce, and G. McCarthy, "Social perception from visual cues: role of the sts region," *Trends in Cognitive Sciences*, vol. 4, no. 7, pp. 267–278, 2000.

[17] N. G. Kanwisher, "Domain specificity in face perception," *Nature Neuroscience*, vol. 3, pp. 759–763, 2000.

[18] N. G. Kanwisher, "Functional specificity in the human brain: A window into the functional architecture of the mind," *Proceedings of the National Academy of Sciences*, vol. 107, pp. 11163 – 11170, 2010.

[19] F. Simion and E. D. Giorgio, "Face perception and processing in early infancy: inborn predispositions and developmental changes," *Frontiers in Psychology*, vol. 6, 2015.

[20] N. G. Kanwisher, J. McDermott, and M. M. Chun, "The fusiform face area: A module in human extrastriate cortex specialized for face perception," *The Journal of Neuroscience*, vol. 17, pp. 4302 – 4311, 1997.

[21] N. G. Kanwisher and G. Yovel, "The fusiform face area: a cortical region specialized for the perception of faces," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 361, pp. 2109 – 2128, 2006.

[22] V. Bruce and A. Young, "Understanding face recognition," *British journal of psychology*, vol. 77, no. 3, pp. 305–327, 1986.

[23] M. J. Farah, K. D. Wilson, H. M. Drain, and J. R. Tanaka, "The inverted face inversion effect in prosopagnosia: Evidence for mandatory, face-specific perceptual mechanisms," *Vision research*, vol. 35, no. 14, pp. 2089–2093, 1995.

[24] D. Y. Tsao and M. S. Livingstone, "Mechanisms of face perception," *Annual review of neuroscience*, vol. 31, p. 411, 2008.

[25] J. V. Haxby, E. A. Hoffman, and M. I. Gobbini, "The distributed human neural system for face perception," *Trends in Cognitive Sciences*, vol. 4, pp. 223–233, 2000.

[26] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4. Springer, 2006.

[27] T. Hastie, R. Tibshirani, and J. Friedman, "Unsupervised learning," in *The elements of statistical learning*, pp. 485–585, Springer, 2009.

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[29] S. Mullainathan and J. Spiess, "Machine learning: an applied econometric approach," *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 87–106, 2017.

[30] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[32] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[33] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[34] J. MacQueen, "Classification and analysis of multivariate observations," in *5th Berkeley Symp. Math. Statist. Probability*, pp. 281–297, 1967.

[35] H. Hotelling, "Analysis of a complex of statistical variables into principal components.," *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[36] B. Liu, "Sentiment analysis and opinion mining," *Synthesis lectures on human language technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[37] J. Shlens, "A tutorial on principal component analysis," *arXiv preprint arXiv:1404.1100*, 2014.

[38] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pp. 649–666, Springer, 2016.

[39] N. Komodakis and S. Gidaris, "Unsupervised representation learning by predicting image rotations," in *International conference on learning representations (ICLR)*, 2018.

[40] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press, 2018.

[41] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[42] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3, pp. 279–292, 1992.

[43] G. A. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*, vol. 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.

[44] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[45] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, “End to end learning for self-driving cars,” *arXiv preprint arXiv:1604.07316*, 2016.

[46] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[47] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, “Self-taught learning: Transfer learning from unlabeled data,” *Proceedings of the 24th international conference on Machine learning*, pp. 759–766, 2007.

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

[49] X. Glorot, A. Bordes, and Y. Bengio, “Domain adaptation for large-scale sentiment classification: A deep learning approach,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 4, p. 27, 2011.

[50] G. Hinton, O. Vinyals, J. Dean, *et al.*, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015.

[51] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

[52] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *International conference on machine learning*, pp. 1126–1135, PMLR, 2017.

[53] R. Caruana, “Multitask learning,” *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[54] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, *et al.*, “Palm: Scaling language modeling with pathways,” *arXiv preprint arXiv:2204.02311*, 2022.

[55] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, *et al.*, “A generalist agent,” *arXiv preprint arXiv:2205.06175*, 2022.

[56] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, *et al.*, “Rt-1: Robotics transformer for real-world control at scale,” *arXiv preprint arXiv:2212.06817*, 2022.

[57] M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” in *Psychology of learning and motivation*, vol. 24, pp. 109–165, Elsevier, 1989.

[58] R. Ratcliff, “Connectionist models of recognition memory: constraints imposed by learning and forgetting functions.,” *Psychological review*, vol. 97, no. 2, p. 285, 1990.

[59] J. Cichon and W.-B. Gan, “Branch-specific dendritic ca2+ spikes cause persistent synaptic plasticity,” *Nature*, vol. 520, no. 7546, pp. 180–185, 2015.

[60] M. Long and J. Wang, “Learning multiple tasks with deep relationship networks,” *ArXiv*, vol. abs/1506.02117, 2015.

[61] Y. Lu, A. Kumar, S. Zhai, Y. Cheng, T. Javidi, and R. S. Feris, “Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1131–1140, 2016.

[62] A. Søgaard and Y. Goldberg, “Deep multi-task learning with low level tasks supervised at lower layers,” *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2016.

[63] K. Hashimoto, C. Xiong, Y. Tsuruoka, and R. Socher, “A joint many-task model: Growing a neural network for multiple nlp tasks,” *ArXiv*, vol. abs/1611.01587, 2016.

[64] S. Liu, E. Johns, and A. J. Davison, “End-to-end multi-task learning with attention,” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1871–1880, 2018.

[65] I. Misra, A. Shrivastava, A. K. Gupta, and M. Hebert, “Cross-stitch networks for multi-task learning,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3994–4003, 2016.

[66] L. Duong, T. Cohn, S. Bird, and P. Cook, “Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser,” in *Annual Meeting of the Association for Computational Linguistics*, 2015.

[67] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi, “Modeling task relationships in multi-task learning with multi-gate mixture-of-experts,” *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018.

[68] Z. Zhao, L. Hong, L. Wei, J. Chen, A. Nath, S. Andrews, A. Kumthekar, M. Sathiamoorthy, X. Yi, and E. H. Chi, “Recommending what video to watch next: a multitask ranking system,” *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019.

[69] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, “Learning in nonstationary environments: A survey,” *IEEE Computational Intelligence Magazine*, vol. 10, no. 4, pp. 12–25, 2015.

[70] R. Hadsell, D. Rao, A. A. Rusu, and R. Pascanu, “Embracing change: Continual learning in deep neural networks,” *Trends in cognitive sciences*, vol. 24, no. 12, pp. 1028–1040, 2020.

[71] H. Qu, H. Rahmani, L. Xu, B. Williams, and J. Liu, “Recent advances of continual learning in computer vision: An overview,” *arXiv preprint arXiv:2109.11369*, 2021.

[72] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, “A continual learning survey: Defying forgetting in classification tasks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.

[73] G. A. Carpenter and S. Grossberg, “A massively parallel architecture for a self-organizing neural pattern recognition machine,” *Comput. Vis. Graph. Image Process.*, vol. 37, pp. 54–115, 1988.

[74] N. D. Daw, Y. Niv, and P. Dayan, “Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control,” *Nature neuroscience*, vol. 8, no. 12, pp. 1704–1711, 2005.

[75] G. Kempermann, D. Gast, and F. H. Gage, “Neuroplasticity in old age: sustained fivefold induction of hippocampal neurogenesis by long-term environmental enrichment,” *Annals of neurology*, vol. 52, no. 2, pp. 135–143, 2002.

[76] A. Kumar, “Long-term potentiation at ca3–ca1 hippocampal synapses with special emphasis on aging, disease, and stress,” *Frontiers in aging neuroscience*, vol. 3, p. 7, 2011.

[77] W. C. Abraham and M. F. Bear, “Metaplasticity: the plasticity of synaptic plasticity,” *Trends in Neurosciences*, vol. 19, no. 4, pp. 126–130, 1996.

[78] D. O. Hebb, *The organization of behavior.* Wiley Inc., New-York, 1949.

[79] M. Benna and S. Fusi, “Computational principles of synaptic memory consolidation,” *Nature Neuroscience*, vol. 19, 10 2016.

[80] G. Parisi, R. Kemker, J. Part, C. Kanan, and S. Wermter, “Continual lifelong learning with neural networks: A review,” *Neural Networks*, vol. 113, pp. 54–71, 02 2019.

[81] Z. Li and D. Hoiem, “Learning without forgetting,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, p. 2935–2947, dec 2018.

[82] R. Aljundi, P. Chakravarty, and T. Tuytelaars, “Expert gate: Lifelong learning with a network of experts,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7120–7129, 2016.

[83] A. Triki, R. Aljundi, M. B. Blaschko, and T. Tuytelaars, “Encoder based lifelong learning,” *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1329–1337, 2017.

[84] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.*, “Overcoming catastrophic forgetting in neural networks,” *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[85] F. Zenke, B. Poole, and S. Ganguli, “Continual learning through synaptic intelligence,” *Proceedings of machine learning research*, vol. 70, pp. 3987–3995, 2017.

[86] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, “Memory aware synapses: Learning what (not) to forget,” in *European Conference on Computer Vision*, 2017.

[87] P. S. Eriksson, E. Perfilieva, T. Björk-Eriksson, A.-M. Alborn, C. Nordborg, D. A. Peterson, and F. H. Gage, “Neurogenesis in the adult human hippocampus,” *Nature Medicine*, vol. 4, pp. 1313–1317, 1998.

[88] H. van Praag, B. R. Christie, T. J. Sejnowski, and F. H. Gage, “Running enhances neurogenesis, learning, and long-term potentiation in mice.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 96 23, pp. 13427–31, 1999.

[89] B. Leuner and E. Gould, “Structural plasticity and hippocampal function.,” *Annual review of psychology*, vol. 61, pp. 111–40, C1–3, 2010.

[90] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, “Progressive neural networks,” *ArXiv*, vol. abs/1606.04671, 2016.

[91] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," *ArXiv*, vol. abs/1708.01547, 2017.

[92] J. Xu and Z. Zhu, "Reinforced continual learning," *ArXiv*, vol. abs/1805.12369, 2018.

[93] X. Li, Y. Zhou, T. Wu, R. Socher, and C. Xiong, "Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting," in *International Conference on Machine Learning*, 2019.

[94] S. C. Y. Hung, J.-H. Lee, T. S. T. Wan, C.-H. Chen, Y.-M. Chan, and C.-S. Chen, "Increasingly packing multiple facial-informatics modules in a unified deep-learning model via lifelong learning," *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, 2019.

[95] S. C. Y. Hung, C.-H. Tu, C.-E. Wu, C.-H. Chen, Y.-M. Chan, and C.-S. Chen, "Compacting, picking and growing for unforgetting continual learning," in *Neural Information Processing Systems*, 2019.

[96] C. Fernando, D. S. Banarse, C. Blundell, Y. Zwols, D. R. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra, "Pathnet: Evolution channels gradient descent in super neural networks," *ArXiv*, vol. abs/1701.08734, 2017.

[97] J. Serrà, D. Surís, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," in *International Conference on Machine Learning*, 2018.

[98] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2017.

[99] R. C. O'Reilly, R. Bhattacharyya, M. D. Howard, and N. A. Ketz, "Complementary learning systems," *Cognitive science*, vol. 38 6, pp. 1229–48, 2014.

[100] L. R. Squire, "Memory and the hippocampus: a synthesis from findings with rats, monkeys, and humans.," *Psychological review*, vol. 99 2, pp. 195–231, 1992.

[101] L. Nadel and M. Moscovitch, "Memory consolidation, retrograde amnesia and the hippocampal complex," *Current Opinion in Neurobiology*, vol. 7, pp. 217–227, 1997.

[102] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory.," *Psychological review*, vol. 102, no. 3, p. 419, 1995.

[103] S. Corkin, "What's new with the amnesic patient h.m.?," *Nature Reviews Neuroscience*, vol. 3, pp. 153–160, 2002.

[104] A. V. Robins, "Catastrophic forgetting, rehearsal and pseudorehearsal," *Connect. Sci.*, vol. 7, pp. 123–146, 1995.

[105] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5533–5542, 2016.

[106] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, “Gradient based sample selection for online continual learning,” in *Neural Information Processing Systems*, 2019.

[107] D. Lopez-Paz and M. Ranzato, “Gradient episodic memory for continual learning,” in *NIPS*, 2017.

[108] Y. Liu, A. Liu, Y. Su, B. Schiele, and Q. Sun, “Mnemonics training: Multi-class incremental learning without forgetting,” *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12242–12251, 2020.

[109] T. L. Hayes, K. Kafle, R. Shrestha, M. Acharya, and C. Kanan, “Remind your neural network to prevent catastrophic forgetting,” in *European Conference on Computer Vision*, 2019.

[110] H. Jégou, M. Douze, and C. Schmid, “Product quantization for nearest neighbor search,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 117–128, 2011.

[111] L. Pellegrini, G. Graffieti, V. Lomonaco, and D. Maltoni, “Latent replay for real-time continual learning,” *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10203–10209, 2019.

[112] H. Shin, J. K. Lee, J. Kim, and J. Kim, “Continual learning with deep generative replay,” in *NIPS*, 2017.

[113] G. M. van de Ven and A. S. Tolias, “Generative replay with feedback connections as a general strategy for continual learning,” *ArXiv*, vol. abs/1809.10635, 2018.

[114] T. Lesort, H. Caselles-Dupré, M. G. Ortiz, A. Stoian, and D. Filliat, “Generative models from the perspective of continual learning,” *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2018.

[115] R. Kemker and C. Kanan, “Fearnet: Brain-inspired model for incremental learning,” *ArXiv*, vol. abs/1711.10563, 2017.

[116] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, “A convolutional neural network cascade for face detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5325–5334, 2015.

[117] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint face detection and alignment using multitask cascaded convolutional networks,” *IEEE signal processing letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[118] C. Zhang and Z. Zhang, “Improving multiview face detection with multi-task deep convolutional neural networks,” in *IEEE Winter Conference on Applications of Computer Vision*, pp. 1036–1041, IEEE, 2014.

[119] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, “Facial landmark detection by deep multi-task learning,” in *European conference on computer vision*, pp. 94–108, Springer, 2014.

[120] R. Ranjan, V. M. Patel, and R. Chellappa, “Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 1, pp. 121–135, 2017.

[121] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[122] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders, "Segmentation as selective search for object recognition," in *2011 international conference on computer vision*, pp. 1879–1886, IEEE, 2011.

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

[124] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-shot multi-level face localisation in the wild," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5203–5212, 2020.

[125] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Springer, 2016.

[126] D. Qi, W. Tan, Q. Yao, and J. Liu, "Yolo5face: why reinventing a face detector," *arXiv preprint arXiv:2105.12931*, 2021.

[127] Y. Xu, W. Yan, G. Yang, J. Luo, T. Li, and J. He, "Centerface: joint face detection and alignment using face as point," *Scientific Programming*, vol. 2020, 2020.

[128] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6569–6578, 2019.

[129] P. Barros, G. Parisi, and S. Wermter, "A personalized affective memory model for improving emotion recognition," in *International Conference on Machine Learning*, pp. 485–494, PMLR, 2019.

[130] N. Churamani and H. Gunes, "Clifer: Continual learning with imagination for facial expression recognition," in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pp. 322–328, IEEE, 2020.

[131] M. Mainsant, M. Solinas, M. Reyboz, C. Godin, and M. Mermillod, "Dream net: a privacy preserving continual leaming model for face emotion recognition," in *2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pp. 01–08, IEEE, 2021.

[132] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2015.

[133] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525, 2016.

[134] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *ArXiv*, vol. abs/1502.03167, 2015.

[135] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *ArXiv*, vol. abs/1804.02767, 2018.

[136] C.-Y. Wang, H.-Y. M. Liao, I.-H. Yeh, Y.-H. Wu, P.-Y. Chen, and J.-W. Hsieh, “Cspnet: A new backbone that can enhance learning capability of cnn,” *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1571–1580, 2019.

[137] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, pp. 1904–1916, 2014.

[138] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, 2018.

[139] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.

[140] D. Qi, W. Tan, Q. Yao, and J. Liu, “Yolo5face: Why reinventing a face detector,” *ArXiv*, vol. abs/2105.12931, 2021.

[141] Z. Feng, J. Kittler, M. Awais, P. Huber, and X. Wu, “Wing loss for robust facial landmark localisation with convolutional neural networks,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2235–2245, 2017.

[142] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” *ArXiv*, vol. abs/2004.10934, 2020.

[143] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, “Albumentations: Fast and flexible image augmentations,” *Information*, vol. 11, no. 2, 2020.

[144] Y. Xiong, K. Zhu, D. Lin, and X. Tang, “Recognize complex events from static images by fusing deep channels,” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1600–1609, 2015.

[145] C. L. Zitnick and P. Dollár, “Edge boxes: Locating object proposals from edges,” in *European Conference on Computer Vision*, 2014.

[146] A. Krizhevsky, “Learning multiple layers of features from tiny images,” 2009.