On-device Saliency Prediction based on Pseudo Knowledge Distillation

***Abstract*—Saliency prediction models aim to mimic the human visual system's attention process, and the research has made significant progress due to recent advancements in Deep Convolution Neural Networks (DCNN). However, the high memory requirements and intensive computational demands make these approaches less suitable for IoT devices, and there's a need for improved computational efficiency and reduced memory footprint to facilitate distributed IoT intelligence. This paper proposes a Pseudo Knowledge Distillation (PKD) training method for creating a compact real-time saliency prediction model. The proposed method can effectively transfer knowledge from computationally expensive once-for-all (OFA-595) as a single teacher model and a combination of OFA-595 and EfficientNet-B7 as a multi-teacher model to an Early Exit Evolutionary Algorithm network (EEEA-Net) student model by utilising knowledge distillation and pseudo labelling. To examine the performance of PKD, five saliency benchmark datasets are used to demonstrate PKD's improved prediction performance and its reduced inference time without modifying the original student model.**

***Index Terms*—Deep Learning, Pseudo Label, Knowledge Distillation, Saliency Prediction.**

I. Introduction

VISION is an essential sense for human beings, and tracking eye movement can be utilised for understanding human visual behaviour. Humans have an amazing ability to focus attention on certain areas instead of inspecting the complete details for interpreting and understanding the scene. The selective attention mechanism of the human visual system (HVS), when mimicked by a computer through simulation, is called saliency prediction or visual attention prediction. This attention mechanism of HVS plays an important role in various applications, including segmentation [1], surface detection [2], image retargeting [3], and object tracking [4].

Artificial Intelligence (AI) for real-time monitoring and surveillance has recently played an important role in industrial informatics, operating as a facilitator with varying degrees of success and interests. Deployment of Internet of Things (IoT) generates big data, subsequently raises the processing demand on the central server, which represents a bottleneck. The availability of such big industrial data and recent advancements in computational intelligence have resulted in a paradigm shift toward AI utilisation. This paradigm change is the major push towards completely autonomous intelligent systems with lower computational complexity. Therefore, there's a need for innovative algorithms with reduced parameters, complexity and latency to be operated on the IoT devices. To perform visual tasks on such IoT devices require compact model, which can be accomplished by model compression [5]. On-device saliency prediction needs a compact model with low computation cost. For real-time on-device saliency prediction, an efficient visual saliency prediction system is critical. A recent study [6] pre-trained a deep saliency detection model on a cloud server, and the backbone of CNN is dynamically chosen and fine-tuned for a particular IoT application based on the processing capacity of fog devices. To analyse the scene, Gao et al. proposed a saliency detection approach that recognises common things (salient regions) in an image collected from different IoT devices with a relatively high computation cost [7]. Our main motivation for this research is to develop a compact deep learning saliency prediction architecture that is flexible and efficient to be deployed on computationally constrained edge devices such as single-board computers or smartphones.

A saliency prediction model receives an input image and generates an output in the form of fixation distribution (saliency map). The saliency map marks the regions of interest with high intensity to indicate that they are more important than others. In general, these models can be classified into two categories based on the attention process they simulate: bottom-up models and top-down models. Top-down attention models are task specific, while bottom-up attention models are for free-viewing when we do not have any objective or task in our mind. The aim of this paper is to mimic the task-free bottom-up visual attention process by predicting salient regions on natural images.

Early saliency prediction models were biologically inspired [8], [9]. They primarily utilise multi low-level handcraft features for instance colour, orientation, texture and intensity, these models blend features in a heuristics way for computing saliency. However, these models generally fail to incorporate high-level features (contextual information, complex objects, etc.), resulting in creating a considerable gap between human and computation model in terms of the prediction accuracy. The saliency prediction domain has transformed from manual feature extraction to automatic representation learning with the help of deep neural networks. Due to this paradigm shift, deep learning models have achieved substantial improvement over traditional saliency prediction models [10].

Deep learning has recently become a popular strategy in saliency prediction domain, recent saliency prediction models utilise deep learning for delivering reasonable accuracy. Although it improves the gap between ground truth and model prediction but at the expense of higher computational cost. They require a large number of parameters and tremendous computational resources, such high resource requirements are not suitable on IoT devices for edge computing.

The computational demand of saliency prediction inference is one limitation to the widespread use of saliency prediction in various applications. Thus, current saliency prediction models need to evolve towards compact and computationally efficient methods that can be deployed on edge devices. Motivated by

this, a new method is proposed called PKD for predicting on-device saliency. The seminal work on knowledge distillation [11] can be observed in various complex tasks across different domains. However, relatively less research can be seen in saliency prediction domain. Therefore, this paper focuses on:

- The computationally efficient approaches for saliency prediction to achieve near state-of-the-art accuracy at a reduced computational cost.
- The development of the model that has less number of parameters and with good accuracy for saliency prediction models. We evaluate the proposed method against state-of-the-art in terms of accuracy metrics (AUC, NSS, CC, etc.) and theoretical complexity (FLOPS, model parameters and latency).
- To fulfil the requirements of on-device applications, a new method is proposed PKD: Pseudo Knowledge Distillation for on-device saliency. This method provides model performance comparable to many more complex models for saliency prediction by transferring knowledge from a computationally expensive teacher model to a small student model by utilising knowledge distillation and pseudo labelling.
- Various experiments performed on different datasets verify the effectiveness of the proposed method. Compared with state-of-the-art approaches, the proposed method achieves competitive accuracy with smaller model size.

II. Related Works

Saliency prediction models can be categorised into three groups: classic (non-deep) saliency models, deep saliency models and compact saliency models.

A. Classic (Non-deep) saliency models

Generally, traditional saliency prediction models are based on hand-craft or manual features. One of the earliest computational models [9] for saliency prediction was proposed by combining multi-scales low-level features (orientation, colour, and intensity) into a final saliency map for saliency prediction. Following this foundational work, various techniques investigated the similar concept such as a work proposed graph-based visual saliency [8] by using Markov chains to generate saliency maps. In classic saliency prediction models researchers leverage their domain knowledge expertise to construct these features manually, which may fail to replicate the human visual system's response to complicated nature situations.

B. Deep saliency models

A recent trend shows that deep learning has been a popular technique for saliency prediction. The first attempt to predict visual saliency using convolutional neural networks was made in 2014 [12]. They used ensemble of deep networks (eDN), in which they blend information from various layers to train a linear classifier for saliency prediction. Following that, researchers began to use deeper models by introducing a transfer learning approach, a variety of networks based on the pre-trained backbone are proposed, including EML-Net [13], SALICON [10] and SalED [14].

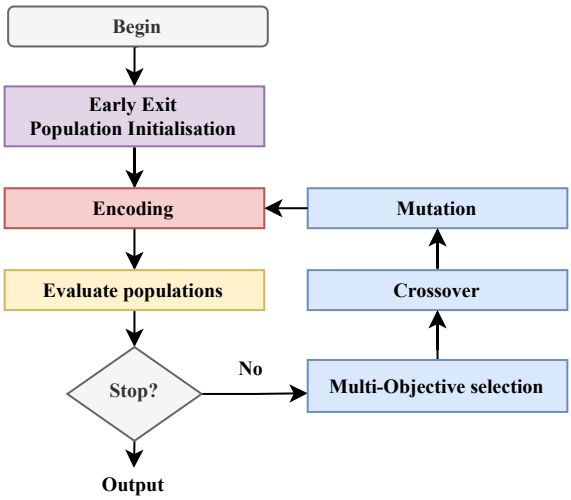

Fig. 1. Block diagram of the Early Exit Evolutionary Algorithm.

C. Compact saliency models

In a computationally constraint environment, fast and compact models are crucial for embedded computer vision applications such as facial verification and image recognition. MobileNetV2 [15] uses depth-wise or separable convolution to reduce FLOPS which helps for on-device deployment. MobileNetV3 [16] is further polished to reduce the computational complexity. GhostNet [17] was proposed to produce additional feature maps from low-cost operations. EfficientNet [18] propose a new scaling technique that evenly scales all dimensions (resolution, width and depth). EEEA-Nets [19] are suitable for on-device processors with constrained computing resources.

Several approaches for the compact and fast saliency prediction model have been proposed in the past. A work [20] reported a faster gaze prediction model, developed by using knowledge distillation and fisher pruning. A recent work [21] modified and utilised different efficient convolution neural networks suitable for inference on constrained computing devices such as MobileNetV2 [15] and EfficientNet [18] for saliency prediction.

D. EEEA-Nets

An algorithm based on Early Exit Population Initialisation (EE-PI) and Evolutionary Algorithm (EA) was proposed to create the EEEA-Net models [19], as shown in Fig. 1. The work utilised the Network Architecture Search (NAS) based method to search for a new model called EEEA-Nets, suitable for on-device processors with constrained computing resources. The aim of the EEEA algorithm is to discover a CNN model that minimises error, FLOPS, and the number of parameters, as shown in Eq. 1, where m is the subnet model and S is the set of subnet models.

$$\begin{aligned} \min \quad & \{Error\,(m)\,, FLOPS\,(m)\,, Params\,(m)\} \\ \text{s.t.} \quad & m \in S \end{aligned} \tag{1}$$

The proposed method utilises EEEA-Net as the student network for hierarchical feature extraction by adapting it for saliency prediction. It is used as an encoder module and there are five stages in encoder module, each stage having four MBConv Blocks [16], for a total of 20 layers in the encoder. The EEEA-Nets model has a similar block structure to the MobileNetV3 model but differs in the size of the filter, number of output channels, number of depth blocks, and input resolution, these values are optimised in EEEA-Net.

There are a few factors that contribute to the selection of EEEA-Net as an encoder in the student network. Since it has been extensively evaluated on a broad range of tasks, including image classification, object detection, image segmentation, and human keypoint detection, and the findings indicate that EEEA-Net-C2 outperforms MobileNetV3. Moreover, EEEA-Net is a subnetwork of the OFA [22] supernet, so it is a viable candidate for selection in our proposed model to achieve performance similar to the OFA-595 subnetwork.

E. Evaluation metrics

Evaluation of different saliency prediction models is based on the consistency between the predicted result and the ground truth. Previous research on saliency evaluation metrics [23] shows that for a fair comparison of the saliency prediction model, a single evaluation metric is not sufficient. Therefore, this paper adopts widely accepted evaluation metrics, including CC, KLD, NSS, SIM, and AUC. These metrics can be divided into two categories: location-based (AUC and NSS) and distribution-based (CC, KLD, and SIM). The difference between these two categories is the way each type uses the ground truth. Location-based metrics treat saliency map values at discrete fixation points, and distribution-based metrics treat both saliency map and ground-truth as a continuous distribution. We denote the predicted saliency map as P, the ground-truth saliency map as G and the ground-truth fixation map as Q.

1) Area Under ROC Curve (AUC): Predicting the fixation location on an image, a saliency map, can be treated as a classifier problem. AUC quantifies the saliency prediction model performance based on fixated and non-fixated locations. Given an image and its ground-truth eye fixation map Q, AUC evaluates the classification performance of the predicted saliency map P. Fixation points and non-fixation points are considered as positive and negative sets, respectively. The computed saliency map P is binary classified into salient and non-salient regions by a threshold. By varying the threshold from 0 to 1, ROC is generated by plotting the true positive vs false positive rate.

2) Normalised Scanpath Saliency (NSS): This evaluation metric is specifically introduced [24] to evaluate the saliency prediction model. It computes the measure of correspondence between saliency maps and ground-truth by taking the mean between model prediction (saliency map) and eye fixations.

$$NSS(P,Q) = \frac{1}{N}\sum_{i} \bar{P}_i \times Q_i \tag{2}$$

where $\bar{P}$ is the unit normalised saliency map P (zero mean and unit standard deviation) and N denotes the total number of human eye fixations.

3) Linear Correlation Coefficient (CC): It is a statistical measure that is utilised to evaluate how closely two random variables are related. For saliency prediction it quantifies the linear correlation between two distributions (model prediction and ground-truth) for the purpose of evaluation. It is computed by using following equation:

$$CC(P,G) = \frac{covar\,(P,G)}{\sigma\,(P) \times \sigma\,(G)} \tag{3}$$

where $covar$ and σ refer to the covariance and standard deviation, respectively. CC score close to $+1$ indicate a perfect linear relationship between the maps.

4) Similarity Metric: SIM (also known as histogram intersection) is a similarity metric [25] that quantifies the similarity between two histogram-based distributions. In saliency domain, it performs a comparison between two saliency maps by calculating the similarity between two distributions. The similarity metric can be computed by using following equation

$$SIM\,(P,G) = \sum_{i} min(P_i, G_i) \tag{4}$$

A SIM of one indicates that the distributions are identical, while a SIM of zero indicates that there is no overlap between distributions.

5) Kullback-Leibler Divergence (KLD): To quantify the statistical distance D between an estimated and a target distribution, KLD is a suitable metric based on information theory. The distribution P is used to approximate the distribution G, KLD evaluates the loss of information, resulting in a probabilistic interpretation of saliency and ground-truth density maps. The equation for KLD is computed by using the following equation:

$$KLD\,(P,Q) = \sum_{i} Q_i log\left(\epsilon + \frac{Q_i}{P_i + \epsilon}\right) \tag{5}$$

where i denotes the i^{th} pixel of an image and ϵ is a regularisation constant. A lower value of KLD shows that the predicted saliency map is more accurate representation of the ground truth.

III. Proposed Method

A. The framework

The proposed PKD method for saliency prediction is shown in Fig. 2. It consists of four major components: teacher network (large), student network (small), pseudo labels and loss function. The proposed method utilises knowledge distillation by training a small model to mimic a larger pre-trained model. The overall model loss is the combination of teacher and student loss. The proposed method utilises pseudo labels with knowledge distillation by providing the student network with the teacher prediction (pseudo labels). Finally, the total loss is calculated by adding teacher loss and student loss.

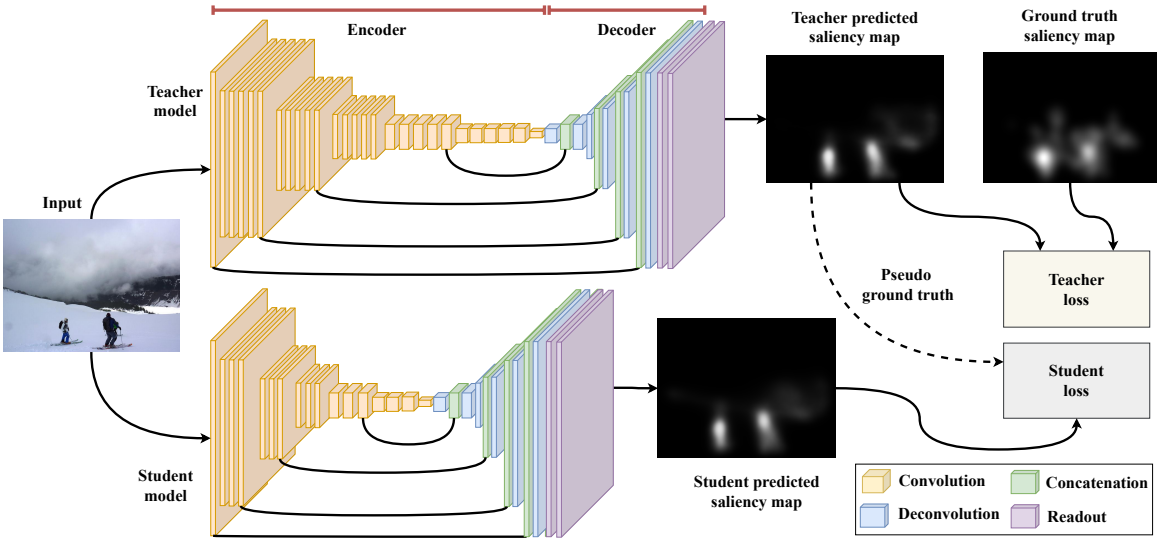

Fig. 2. The overall architecture of the proposed saliency prediction model.

B. Computationally efficient model through knowledge distillation and pseudo labels

Commonly used methods for reducing the size of over-parameterised neural network are pruning [26] and knowledge distillation [11]. Reducing the parameters of neural network by using pruning has some limitations, for instance after pruning the final network has usually similar network structure, which is not desirable for edge devices with constrained computational capacity. On the other hand, knowledge distillation is widely utilised to transfer learned information from one model to another using the student-teacher learning framework. This paper adapts the idea of knowledge distillation to solve saliency prediction problems.

The proposed method utilises knowledge distillation as shown in Fig. 2. OFA-595 [22] is used as a teacher (large) network, while EEEA-Net [19] is used as a student (small) network. For on-device saliency prediction OFA-595 subnetwork from supernet is adapted by utilising it as the backbone (pre-trained with imageNet1k) for the teacher network. In OFA it trains one network and specialise that network for efficient deployment by allowing it to be deployed under diverse architectural configurations. It employs a novel progressive shrinking algorithm that reduces the model size across various dimensions (depth, width, kernel size and resolution). It can generate many subnetworks that are compatible with a variety of hardware platforms and latency limitations while retaining a high level of accuracy. The proposed method adapts the EEEA-Net model to the saliency prediction problem by using it as a backbone (pre-trained with imageNet1k) of the student network. The same decoder and ReadOut module are utilised in both the teacher and student networks. More information on the decoder and ReadOut module can be found in the sections III-D and III-E respectively.

A recent trend shows that pseudo labels have been utilised extensively for computer vision. The proposed method utilises pseudo labels as shown in Fig 3. The training step involves X

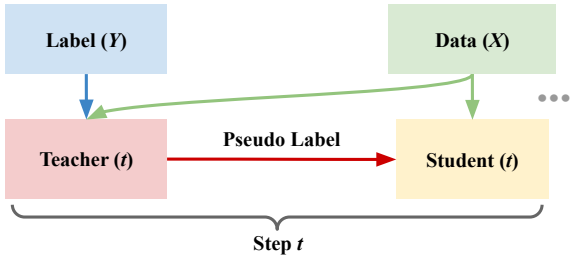

Fig. 3. Pseudo Knowledge Distillation (PKD) method. Each timestamp t of the training process involves few steps, which include X as input data (input image), Y represents a label (saliency map or fixations) and pseudo labels generated by the teacher network are given to the student network for knowledge distillation.

as input data (input image) and Y represent a label (saliency map or fixations). Both the teacher and student networks receive same data (input image) and they generate their respective predictions, however the teacher prediction (saliency map or fixations) is provided to the student network as a pseudo label. This pseudo label is treated as ground truth by the student network, and the student loss is calculated based on student prediction and ground-truth (pseudo label), implying that the student network will learn from the teacher network by using pseudo labels. Further detail of pseudo labels working can be seen in Algorithm 1.

As illustrated in Fig. 2, the proposed teacher and student models receive an input image from the SALICON dataset. Both teacher and student networks will predict their respective saliency map. Let the teacher predicted saliency map be denoted as P_t, ground truth as G and student predicted saliency map as P_s. The teacher loss function will calculate the difference between the teacher's prediction P_t and ground truth G. On the other hand, student loss will be calculated by the loss function with student prediction P_s and pseudo ground truth P_t (teacher's prediction).

The proposed method utilises knowledge distillation to train

Algorithm 1 Pseudo-Knowledge Distillation

Input: Teacher Network T, Student Network S.
Output: Student Network S
P_t is the prediction of teacher network, P_s is the prediction of student network, G is the corresponding ground truth, G_p is the pseudo ground truth, W_{ts} is the weights of teacher and student networks, L is the loss values, D is the dataset.
for each batch in D **do**
  Train T with the input images;
  Calculate teacher loss L_T with P_t, G; (see Eq. 8)
  $G_p \leftarrow P_t$;
  Train S with the input images;
  Calculate student loss L_S with P_s, G_p; (see Eq. 8)
  Calculate L_{total} with L_T, L_S; (see Eq. 7)
  Backpropagation T and S by L_{total}
end for
Evaluate T and S on D
Return S

the student network. The total loss L_{total} is the weighted sum of teacher loss and student loss.

$$L_{total} = W_{ts} L_T\left(P_t, G\right) + \left(1 - W_{ts}\right) L_S\left(P_s, P_t\right) \quad (6)$$

where P_t is the predicted saliency map of the teacher network, the proposed method utilise P_t in teacher and student loss functions. In the teacher loss function it is utilised as teacher prediction, however in the student loss function, P_t is used as a pseudo label. Where $L(\cdot)$ represents the combined loss function.

Weights are fixed for the proposed method, after performing various experiments it is observed that weights value $W_{ts} = 0.5$ is the optimal weight for teaching the student network (EEEA-Net-C2) by a teacher network (OFA-595). The loss formula at $W_{ts} = 0.5$ can be formulated as:

$$L_{total} = L_T\left(P_t, G\right) + L_S\left(P_s, P_t\right) \quad (7)$$

where L_T is teacher loss and L_S is student loss. For more detail, see Algorithm 1.

C. Loss function

In the literature, many evaluation metrics have been proposed for saliency prediction, as described in Section II-E. In the proposed method, evaluation metrics are used as a loss function for both teacher and student networks. Recently, some work demonstrates that combining loss functions can improve model efficiency [13], [14]. It is due to the fact that each saliency assessment metrics has a distinct emphasis, a single loss may not be adequate to thoroughly acquire saliency information. By testing various loss combinations, it was learnt that the combination of $KLD + CC + AUC$ gives the best performance. Thus, a loss combination is utilised for the proposed method.

$$Loss\left(P, G\right) = KLD\left(P, G\right) + CC\left(P, G\right) + AUC \quad (8)$$

where P and G indicate the predicted saliency map and the ground truth, respectively.

D. A simple and compact decoder

After the encoder output, a decoder network is required for saliency map generation, a simple decoder [27] is used in both teacher and student networks, for simple and computationally efficient output generation. Starting at the last layer of the encoder, the decoder network symmetrically increases the spatial resolution of features for accurate localisation. Our decoder expansion can be divided into different stages for recovering the target resolution, the stage-wise spatial resolution enhancement is given as $1/16$, $1/8$, $1/4$, $1/2$, 1. Each stage of the decoder expansion involves concatenation from the corresponding encoder layer, convolution, activation, and upsampling. Finally, the output is then resized to the input size by using bilinear upsampling.

E. ReadOut module

Generally, in the ReadOut module [27], there are few convolution layers along with activation functions, followed by 1×1 convolution to adjust the size of the output saliency map, making the model simpler and more effective. The proposed model's readout architecture comprises two 3×3 convolutional layers; the first convolution layer is followed by ReLU activation function, while the second utilises sigmoid activation function to produce the saliency map.

IV. Experiments and Results

A. Datasets

Five widely used datasets are utilised for model evaluation, including SALICON, CAT2000, MIT1003, OSIE, and PASCAL-S. SALICON is the largest publicly available dataset for saliency prediction [10], [13], [28], including 10,000 training images, 5,000 validation images, and 5,000 testing images. CAT2000 has 2,000 training and 2,000 testing images, MIT1003 have 1,003 images, OSIE contains 700 images, and PASCAL-S consists of 700 images.

B. Implementation details

Our proposed method is implemented using PyTorch. For training, teacher and student network backbones are initialised with the ImageNet1k weights. The decoder module weights are initialised randomly with the default setting of PyTorch. Adam optimiser is used for both the teacher and student networks. Finally, the model is trained using the SALICON, CAT2000, MIT1003, OSIE, and PASCAL-S training set and monitored convergence using the respective validation set. The model is first trained on the SALICON dataset before being fine-tuned on other datasets.

Each dataset is divided into training and validation sets, with 10,000 training images and 5,000 validation images for the SALICON dataset, 1,600 training images and 400 validation images for the CAT2000 dataset, 800 training images and 200 validation images for the MIT1003 dataset, 500 training images and 200 validation images for OSIE dataset and 650

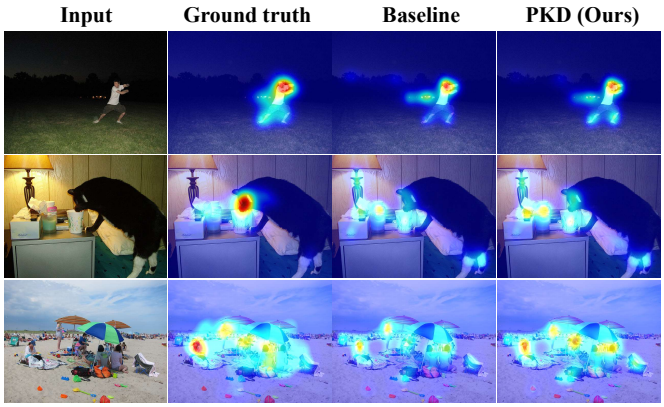

Fig. 4. Distribution of human fixation points. The baseline utilises EEEA-Net-C2 without knowledge distillation, and Ours employs Pseudo Knowledge Distillation on EEEA-Net-C2.

TABLE I
RESULT OF OUR MODEL ON SALICON VALIDATION DATASET WITH COMBINATION LOSS. ARROWS INDICATE THE ASSESSMENT OF SIMILARITY (↑) OR DISSIMILARITY (↓) BETWEEN PREDICTIONS AND TARGETS. THE ENTRIES IN BOLD INDICATE THE BEST RESULTS.

Teacher	CC ↑	KLD ↓	NSS ↑	SIM ↑	AUC ↑
-	0.9007	0.2021	1.9203	0.7932	0.8544
TResNet-M [29]	0.9056	0.1922	1.9320	0.7967	0.8554
OFA-595 [22]	**0.9062**	**0.1907**	1.9298	**0.7987**	**0.8559**
MobileNetV3 [16]	0.9034	0.1965	1.9210	0.7939	0.8550
EfficientNet-B0 [18]	0.9041	0.1937	1.9265	0.7956	0.8549
PNASNet-5 [30]	0.9044	0.1956	1.9319	0.7968	0.8552
VGG-16 [28]	0.8949	0.2131	1.9119	0.7833	0.8535
EfficientNet-B4 [18]	0.9055	0.1924	**1.9346**	0.7980	0.8550

training images and 200 validation images for PASCAL-S dataset. The input image resolution is maintained at 384×384 for all datasets. All the experiments are conducted using four NVIDIA V100 GPUs (32 GB), and Adam optimiser is utilised.

C. Performance analysis of various teacher models

Table I depicts the performance of PKD teacher and PKD student, state-of-the-art networks are used as PKD teacher network to teach PKD student network, the PKD teacher networks used in the experiments are TResNet [29], OFA-595 [22], MobileNetV3 [16], EfficientNet [18], PNASNet [30], and VGG [28]. While preserving high-performance criteria and minimal computational cost, the proposed method effectively transferred knowledge from teacher to student. Experimental results show that OFA-595 is the best PKD teacher network for teaching the proposed PKD student network (EEEA-Net-C2), with increased performance across four metrics, including CC, KL, SIM, and AUC. Experimental results demonstrate that PKD method is not only accurate but also competitive in terms of computational efficiency. Fig. 4 shows sample output of fixation distribution (saliency map).

D. Ablation analysis

This section explains the ablation analysis by showing results on the SALICON validation dataset. We measure the efficiency of the proposed PKD model by determining how well the knowledge is transferred to the student by teacher network by utilising pseudo knowledge distillation. Table II summarises all the findings for ablation analysis. In experiments OFA-595 is used as a teacher network to train various student models. From the results, it is found that when the PKD teacher network is used, the model achieves continuously improved progress across all five metrics, which demonstrates the effectiveness of our proposed method.

TABLE II
MODEL ABLATION ANALYSIS FOR THE SALICON VALIDATION DATASET. THE BASELINE STUDENT MODELS DO NOT USE KD, WHEREAS THE MODELS LABELLED WITH † USE OFA-595 AS THE PKD TEACHER.

Student	CC ↑	KLD ↓	NSS ↑	SIM ↑	AUC ↑
EEEA-Net-C1 [19]	0.8930	0.2213	1.9120	0.7853	0.8522
EEEA-Net-C1†	**0.8995**	**0.2015**	**1.9168**	**0.7908**	**0.8546**
EEEA-Net-C2 [19]	0.8978	0.2032	1.9233	0.7889	0.8548
EEEA-Net-C2†	**0.9062**	**0.1907**	**1.9298**	**0.7987**	**0.8559**
MobileNetV2 [15]	0.8859	0.2254	1.8995	0.7769	0.8518
MobileNetV2†	**0.8915**	**0.2176**	**1.9030**	**0.7808**	**0.8540**
MobileNetV3 [16]	0.9018	0.2030	1.9250	**0.7948**	0.8540
MobileNetV3†	**0.9024**	**0.1987**	**1.9293**	0.7941	**0.8557**
GhostNet [17]	0.8963	0.2087	1.9107	0.7885	0.8534
GhostNet†	**0.9014**	**0.1994**	**1.9189**	**0.7926**	**0.8547**
EfficientNet-B0 [18]	0.9008	0.2173	1.9296	0.7946	0.8538
EfficientNet-B0†	**0.9061**	**0.1950**	**1.9347**	**0.7982**	**0.8560**

TABLE III
LOSS COMBINATION ON SALICON VALIDATION DATASET WITH EEEA-NET-C2 AS BASELINE.

Combination	CC↑	KL↓	NSS↑	SIM↑	AUC↑
CC+KL	**0.8960**	0.2101	1.9142	0.7883	0.8635
CC+AUC	**0.8960**	1.0687	**1.9175**	0.7809	0.8632
SIM+AUC	0.8947	1.0502	1.9101	0.7880	0.8611
CC+KL+SIM	0.8956	0.2154	1.9157	**0.7886**	0.8635
CC+KL+AUC	**0.8960**	**0.2099**	1.9141	0.7883	**0.8636**

1) Combining loss function: Combining evaluation metrics show improved performance compared to when single metric is utilised. Table III illustrates the combination of various evaluation metrics and its performance on the EEEA-Net-C2 model. There are two sorts of combinations considered: two metrics combinations and three metrics combinations. it is learnt that the EEEA-Net-C2 model produces best results when $KLD+CC+AUC$ metrics are combined. All the summarised results are generated by combining loss functions.

2) Multi-teacher learning: Multiple teacher model training may be used to increase the efficacy of knowledge distillation employing teacher student training for developing accurate and compact neural network. We used a multi-teacher approach to distil knowledge from two teacher models to a single compact student model EEEA-Net-C2. Table IV shows the effectiveness of multi-teacher training approach, the multi-teacher training method yields an increase in performance, for instance when EfficientNet-B7 and OFA-595 are used as multi-teacher to train compact student model EEEA-Net-C2 an improvement across three metrics is observed.

Although multi-teacher training improves accuracy across metrics, it comes at a cost of high training time. Table IV compares the training time of single teacher and multi-teacher PKD. When compared to the baseline, single teacher PKD utilises 1.9 to 2.8 times more training time, while multi-teacher PKD consume 2.8 to 3.9 times higher training time.

TABLE IV
RESULT OF MULTI-TEACHER METHOD FOR STUDENT (EEEA-NET-C2) WITH TRAINING TIME ON SALICON VALIDATION DATASET WITH COMBINATION LOSS. ARROWS INDICATE THE ASSESSMENT OF SIMILARITY (↑) OR DISSIMILARITY (↓) BETWEEN PREDICTIONS AND TARGETS. THE ENTRIES IN BOLD INDICATE THE BEST RESULTS

Method	Teacher	CC↑	KL↓	NSS↑	SIM↑	AUC↑	Time (min)	Speedup
Baseline	-	0.9007	0.2021	1.9203	0.7932	0.8544	24	1×
Single Teacher PKD	OFA-595 [22]	0.9062	0.1907	1.9298	0.7987	0.8559	45	1.9×
Single Teacher PKD	EfficientNet-B4 [18]	0.9055	0.1924	1.9346	0.7980	0.8550	50	2.1×
Single Teacher PKD	PNASNet-5 [30]	0.9044	0.1956	1.9319	0.7968	0.8552	68	2.8×
Multi-Teacher PKD	EfficientNet-B4 [18] + OFA-595 [22]	0.9079	**0.1879**	1.9360	0.7998	0.8558	68	2.8×
Multi-Teacher PKD	PNASNet [30] + OFA-595 [22]	0.9072	0.1896	1.9367	0.7994	**0.8566**	88	3.7×
Multi-Teacher PKD	PNASNet [30] + EfficientNet-B4 [18]	0.9068	0.1912	**1.9372**	0.7989	0.8556	93	3.9×
Multi-Teacher PKD	EfficientNet-B7 [18] + OFA-595 [22]	**0.9080**	**0.1879**	1.9367	**0.8002**	0.8555	78	3.3×

TABLE V
COMPARISON PERFORMANCE OF TEACHER (LR_T) AND STUDENT (LR_S) LEARNING RATES BY UTILISING OFA-595 AS TEACHER AND EEEA-NET-C2 AS STUDENT ON SALICON VALIDATION DATASET.

LR_T	LR_S	CC↑	KL↓	NSS↑	SIM↑	AUC↑
0.001	0.001	0.9052	0.1957	**1.9360**	**0.7982**	0.8555
0.0001	0.0001	0.8969	0.2075	1.9097	0.7883	0.8538
0.0001	0.001	**0.9057**	**0.1916**	1.9317	0.7975	**0.8561**
0.001	0.0001	0.8975	0.2065	1.9131	0.7891	0.8539

TABLE VI
COMPARISON PERFORMANCE OF W_{ts} OF TEACHER AND STUDENT LOSSES (EQ.6) FOR SALICON VALIDATION DATASET BY USING PKD (OFA-595 AS TEACHER AND EEEA-NET-C2 AS STUDENT).

W_{ts}	CC↑	KL↓	NSS↑	SIM↑	AUC↑
0.1	0.9058	0.1914	1.9332	0.7980	0.8558
0.2	0.9057	0.1915	1.9319	0.7977	**0.8564**
0.3	0.9058	0.1914	1.9333	0.7980	0.8561
0.4	**0.9059**	0.1915	**1.9336**	**0.7981**	0.8562
0.5	**0.9059**	**0.1913**	1.9326	0.7979	0.8561
0.6	**0.9059**	0.1919	1.9323	0.7977	0.8562
0.7	0.9058	**0.1913**	1.9330	0.7979	0.8561
0.8	0.9051	0.1928	1.9319	0.7970	0.8562
0.9	0.9049	0.1927	1.9295	0.7967	0.8558

3) Comparison performance of learning rate and weights of teacher and student models: Experimental results support the reason for keeping a different learning rate for the teacher and student models; various combinations of teacher and student learning rates are examined and the best one is selected among several combinations. Table V compares the performance of various learning rates combinations for teacher (OFA-595) and student (EEEA-Net-C2) models.

Model averaging is an approach with multiple models contributing toward the final prediction. Weight averaging involves weighted sum of multiple models to predict the outcome. The model weights are small positive values, and the sum of all weights values is equals to 1, indicating that the weights represent the proportion of predicted performance from each model. Table VI show different weights W_{ts} and their performance across various evaluation metrics, the calculation is based on Eq. 6 with OFA-595 as the teacher and EEEA-Net-C2 as the student model on the SALICON dataset, the result indicates that weight values from 0.4-0.6 have better results.

E. Architecture Transfer

Additionally, quantitative analysis is performed on other datasets such as CAT2000, MIT1003, OSIE, and PASCAL-S, and the results are summarised in Table VII. The findings indicate that the combination of OFA-595 and EfficientNet-B7 is an effective multi-teacher since CAT2000 increases accuracy across three metrics, MIT1003 increases accuracy across four, OSIE increases accuracy across five, and PASCAL-S increases accuracy across four.

TABLE VII
RESULT ON FOUR VALIDATION DATASETS WITH COMBINATION LOSS. † REPRESENTS MULTI-TEACHER PKD (OFA-595 AND EFFICIENTNET-B7) TRAINING EEEA-NET-C2 NETWORK AND BASELINE ONLY UTILISE EEE-NET-C2.

Dataset	CC ↑	KL ↓	NSS ↑	SIM ↑	AUC ↑
CAT2000	0.8644	0.3082	**2.4230**	0.7394	**0.8958**
CAT2000 †	**0.8715**	**0.2964**	2.4161	**0.7450**	0.8956
MIT1003	0.7574	0.5998	2.6836	0.6167	**0.8928**
MIT1003 †	**0.7686**	**0.5732**	**2.7450**	**0.6246**	0.8878
OSIE	0.8097	0.5078	3.4447	0.6530	0.9437
OSIE †	**0.8260**	**0.4766**	**3.4713**	**0.6555**	**0.9466**
PASCAL-S	0.8314	0.4490	2.8201	0.6759	**0.9053**
PASCAL-S †	**0.8383**	**0.4395**	**2.8589**	**0.6802**	0.8979

F. On-device evaluation

Table VIII shows the computational performance of the PKD student models. The experiments are conducted using Intel Core i7-6700HQ CPU, running the student model 100 times, and taking the average values. Table VIII shows that the proposed PKD student model (EEEA-Net-C1) has the lowest average latency among all models. Along with the latency other on-device evaluation assessment measures are also considered to evaluate run time performance, such as computational complexity (FLOPS), parameters (Params), model size and memory usage. Findings indicate that EEEA-Net-C1 is a competitive model compared to other state-of-the-art models by only having marginal difference. For instance, EEEA-Net-C1 has a marginal increase in the computational complexity (FLOPS), parameters, memory usage and model size which is 9.1%, 6.7%, 1.1% and 6.1% compared to GhostNet and MobileNetV2 respectively. Overall, the findings indicate that EEEA-Net-C1 is a promising candidate model for on-device edge computing applications.

G. Limitation

One limitation of the proposed work is that knowledge distillation in teacher student framework requires a significant amount of processing time and GPU memory during the

TABLE VIII
ON-DEVICE RESULTS FOR PKD STUDENT MODELS. WHERE LAT REPRESENTS LATENCY.

Model	FLOPS (G)	Params (M)	Memory (MB)	Size (MB)	Lat (ms)
VGG16 [28]	60.81	24.93	250	87.0	1411
OFA-595 [22]	3.29	7.41	200	28.6	371
EEEA-Net-C1 [19]	1.44	3.64	94	14.0	**211**
EEEA-Net-C2 [19]	1.77	4.57	134	17.6	301
MobileNetV2 [15]	2.76	**3.41**	164	**13.2**	266
MobileNetV3 [16]	1.56	4.05	113	15.6	**211**
GhostNet [17]	**1.32**	3.75	**93**	14.5	247
EfficientNet-B0 [18]	3.73	5.44	202	21.0	407

training process. The second limitation is the selection criteria of the teacher network, which is done manually, it may be replaced with an automated method in the future work. Finally, scalability of the proposed model with increased dataset size can be investigated in the future.

V. CONCLUSION

This paper proposes PKD for on-device saliency prediction, a new computationally efficient model suitable for on-device platforms. Traditional saliency prediction models suffer from large parameters and high floating-point operations, which makes them not suitable for hardware constrained real-time applications. We utilised knowledge distillation and pseudo labelling technique for knowledge transfer from a single teacher (OFA-595) and multi-teacher (OFA-595 and EfficientNet-B7) to a compact student model (EEEA-Net). Furthermore, we introduce the loss combination technique for improving the accuracy of the proposed model. Experimental results show that the proposed PKD algorithm achieves the best saliency prediction accuracy in 84% of the experiments over five validation datasets and five evaluation metrics. The EEEA-Net student network also achieves minimal latency (211 ms) with comparable FLOPS and number of parameters against seven benchmark student models. We conclude that PKD is an effective technique for saliency prediction on computationally constrained devices.

REFERENCES

[1] Y. Zhao, J. Zhao, J. Yang, Y. Liu, Y. Zhao, Y. Zheng, L. Xia, and Y. Wang, "Saliency driven vasculature segmentation with infinite perimeter active contour model," *Neurocomputing*, vol. 259, pp. 201–209, 2017.

[2] X. Zhou, Y. Wang, Q. Zhu, J. Mao, C. Xiao, X. Lu, and H. Zhang, "A surface defect detection framework for glass bottle bottom using visual attention model and wavelet transform," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2189–2201, 2019.

[3] Z. Chen, J. Lin, N. Liao, and C. W. Chen, "Full reference quality assessment for image retargeting based on natural scene statistics modeling and bi-directional saliency similarity," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5138–5148, 2017.

[4] S. C. Wong, V. Stamatescu, A. Gatt, D. Kearney, I. Lee, and M. D. McDonnell, "Track everything: Limiting prior knowledge in online multi-object recognition," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4669–4683, 2017.

[5] S. Fu, Z. Li, K. Liu, S. Din, M. Imran, and X. Yang, "Model compression for iot applications in industry 4.0 via multiscale knowledge transfer," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6013–6022, 2019.

[6] C. Wang, S. Dong, X. Zhao, G. Papanastasiou, H. Zhang, and G. Yang, "SaliencyGAN: Deep Learning Semisupervised Salient Object Detection in the Fog of IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2667–2676, 2020.

[7] Z. Gao, C. Xu, H. Zhang, S. Li, and V. H. C. de Albuquerque, "Trustful Internet of Surveillance Things Based on Deeply Represented Visual Co-Saliency Detection," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4092–4100, 2020.

[8] J. Harel, C. Koch, and P. Perona, "Graph-Based Visual Saliency," in *Advances in Neural Information Processing Systems 19*, vol. 19, 2006, pp. 545–552.

[9] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[10] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "SALICON: Saliency in Context," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1072–1080.

[11] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *arXiv preprint arXiv:1503.02531*, 2015.

[12] E. Vig, M. Dorr, and D. Cox, "Large-Scale Optimization of Hierarchical Features for Saliency Prediction in Natural Images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2798–2805.

[13] S. Jia and N. D. Bruce, "EML-NET:An Expandable Multi-Layer NETwork for Saliency Prediction," *Image and Vision Computing*, vol. 95, p. 103887, 2020.

[14] Z. Wang, Z. Liu, W. Wei, and H. Duan, "SalED: Saliency prediction with a pithy encoder-decoder architecture sensing local and global information," *Image and Vision Computing*, vol. 109, p. 104149, 2021.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[16] A. Howard, R. Pang, H. Adam, Q. Le, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, and Y. Zhu, "Searching for MobileNetV3," in *IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1314–1324.

[17] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More Features From Cheap Operations," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1580–1589.

[18] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *International Conference on Machine Learning*, 2019, pp. 6105–6114.

[19] C. Termritthikun, Y. Jamtsho, J. Ieamsaard, P. Muneesawang, and I. Lee, "EEEA-Net: An Early Exit Evolutionary Neural Architecture Search," *Engineering Applications of Artificial Intelligence*, vol. 104, p. 104397, 2021.

[20] L. Theis, I. Korshunova, A. Tejani, and F. Huszár, "Faster gaze prediction with dense networks and Fisher pruning," *arXiv preprint arXiv:1801.05787*, 2018.

[21] F. Hu and K. McGuinness, "FastSal: a Computationally Efficient Network for Visual Saliency Prediction," in *International Conference on Pattern Recognition (ICPR)*, 2021, pp. 9054–9061.

[22] H. Cai, C. Gan, T. Wang, Z. Zhang, and S. Han, "Once for All: Train One Network and Specialize it for Efficient Deployment," in *Eighth International Conference on Learning Representations*, 2020.

[23] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What Do Different Evaluation Metrics Tell Us About Saliency Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 3, pp. 740–757, 2019.

[24] R. J. Peters, A. Iyer, L. Itti, and C. Koch, "Components of bottom-up gaze allocation in natural images," *Vision Research*, vol. 45, no. 18, pp. 2397–2416, 2005.

[25] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[26] J.-H. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5058–5066.

[27] N. Reddy, S. Jain, P. Yarlagadda, and V. Gandhi, "Tidying Deep Saliency Prediction Architectures," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 10 241–10 247.

[28] A. Kroner, M. Senden, K. Driessens, and R. Goebel, "Contextual encoder-decoder network for visual saliency prediction." *Neural Networks*, vol. 129, pp. 261–270, 2020.

[29] T. Ridnik, H. Lawen, A. Noy, E. Ben, B. G. Sharir, and I. Friedman, "TResNet: High Performance GPU-Dedicated Architecture," in *IEEE Winter Conference on Applications of Computer Vision*, 2021, pp. 1399–1408.

[30] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. L. Yuille, J. Huang, and K. Murphy, "Progressive Neural Architecture Search," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 19–35.