Shapley Deep Learning: A Consensus for General-Purpose Vision Systems

Youcef Djenouri
University of South-Eastern Norway, Norway
NORCE Norwegian Research Centre, Norway
youcef.djenouri@usn.no, yodj@norceresearch.no

Ahmed Nabil Belbachir
NORCE Norwegian Research Centre
Norway
nabe@norceresearch.no

Tomasz Michalak
IDEAS NCBR, Warsaw, Poland
tomasz.michalak@ideas-ncbr.pl

Anis Yazidi
OsloMet University, Oslo, Norway
anisy@oslomet.no

Abstract

Shapely Deep Learning (SDL) targets a new foundation for the design of general-purpose vision systems, by establishing a consensus method that facilitates self-adaptation and flexibility to deal with new computer vision tasks. Today, machine learning did not yet reach the flexible, general-purpose intelligence that biological vision has in mimicking visual descriptions and learning in general-purpose vision algorithms. Currently, each model is built using the domain knowledge of the application in question. Data scientists must consequently be well-versed in the relevant subject. This paper presents SDL as a consensus method for general-purpose intelligence without the help of a domain expert as the trained model has been developed utilizing a general deep learning approach that investigates the contribution of each model in the training process. First, several deep learning models have been trained for each image. The shapley value is then determined to compute the contribution of each subset of models in the training. The model selection is finally performed based on the shapley value and the joint model cost. Optimization of the shapley computation is also carried out by investigating the banzhaf function. We present the evaluation of the generality of SDL using the computer vision benchmarks: MNIST for Shapley value performance, ImageNet for image classification, and COCO for object detection. The results reveal the effectiveness of SDL in terms of accuracy and competitiveness of inference runtime. Concretely, SDL achieved 10%, and 8% over MViTv2 for classification, and object detection tasks, respectively.

1. Introduction

Today computer vision systems can only cope with conditions they were designed for, with single-purpose models for specific applications [3, 4, 8, 30] or single tasks such as classification [28], segmentation [29], and object detection [49]. Whenever the conditions change, humans need to step in to redesign the system to adapt it to a new task or dataset that may require an architecture change and retraining. In the future hyper-connected digital world, computer vision systems will need to work radically different from today to make use of panoplies of deep learning architectures developed to meet a wide range of practical applications and conditions. Unfortunately, some architectures performs better than others on some data and vise-versa. Ensemble learning have been largely investigated to solve the uncertainty in the model behavior where several models have been performed and aggregated to achieve better performance [25, 46]. Two main challenges have not been addressed yet for ensemble learning. The first challenge is that these models are time and memory consuming, where all models in the ensemble need to be loaded and executed during the inference phase. The second challenge is that one or more models in the ensemble can contribute negatively to the learning process. To address the aforementioned challenges, we need to answer to the two following research questions: 1. Assume we have a set of models which solves a given computer vision task, what are the best models in this set? In other words, can we distinguish between the models that contribute positively and the models that contribute negatively to the learning process?. 2. Assume we succeeded to derive the best models, how can we explore these models to achieve a better performance.? This work strives to elaborate on these research questions by proposing a new consensus method called Shapely Deep Learning (SDL) as a foundation for the design of general-purpose vision systems.

Motivations The Shapley value is a concept in cooperative game theory that provides a way to fairly distribute

the benefits of a cooperative effort among a group of players based on their individual contributions to the overall effort. The idea behind the Shapley value is that each player's contribution to the overall value of the group should be proportional to the marginal contribution they make to the group's value. The contribution of a player is determined by the change in the value of the group that results from the player's participation [43]. Motivated by the success of Shapley value in addressing different machine learning challenges including feature selection [45], interpretation [36], and data valuation [10], we will elaborate on a consensus method to determine the contribution of each model to the overall value generated by an ensemble models. In this context, each model can be considered as a player in a cooperative game, and the goal is to distribute the total learning process generated by the ensemble model among the models. The Shapley value for model contribution provides a way to assess the relative importance of each model in the learning process. It gives a fair and consistent evaluation of model importance, taking into account not only the direct effect of each model, but also its interaction with other models.

Contributions To the best of our knowledge, this is the first piece of work that thoroughly examines and computes the importance of models using Shapley value to effectively address the challenges of the current computer vision systems. In short, this paper proposes the novel approach SDL as a foundation for designing general-purpose vision systems to tackle the limitations of existing single-purpose models and multi-purpose models such as Mask R-CNN [15] on being tailored to specific tasks. The main contributions of this research work are given as follows:

1. We propose a SDL as a novel baseline for consensus method, which explores the Shapley to first compute the contribution of each model in the set of model players that will be used in the learning process. The information derived by Shapley will be then used to efficiently select the best models for the inference purpose.

2. We introduce two coalition functions to determine the contributions of the models. The loss coalition function that takes directly the loss of each model in the group to assess the coalition value of the group of the models. However, the data coalition takes into account the model output for determining the coalition value of the group of the models.

3. We suggest two different ways of exploiting the selected best models. The first solution employs the average voting without considering the importance of the best models. The second solution considers the weighted voting mechanism, where the weight of each selected model is determined based on its contribution of the set of best models.

4. We conduct extensive experiment to analyze the different components of SDL using three well-established computer vision benchmarks, MNIST for Shapley value performance, ImageNet for image classification, and COCO for object detection, and with different metrics (classification rate, and intersection over union). The results show that SDL outperforms the baseline solutions for both classification and object detection in terms of the quality of the outcomes, and it is very competitive in terms of the inference runtime.

2. Related Work

SDL is a foundation for general-purpose vision systems that considers the benefits of the best models for solving a given computer vision task. Existing works can be roughly grouped into two families, ensemble learning and Shapley learning. In the following, we will give insights of using SDL compared to studies belong to both families.

Ensemble Learning for Computer Vision Ensemble learning for computer vision is a powerful learning paradigm that combines various computer vision models for improving accuracy and robustness of single-based models. It can be divided into four categories: 1) **Bagging** [48, 34, 42]: Bagging or bootstrap aggregating is a technique where multiple copies of a single model are trained on different subsets of the training images. The final output is made by averaging the different outputs of all models. Bagging is particularly useful for reducing overfitting in high-variance models. 2) **Boosting** [47, 40, 13]: It is a technique that combines multiple weak visual learners to create a strong learner. The weak visual learners are trained sequentially, and each new model focuses on the images that were incorrectly trained. Boosting can improve the accuracy of the model and reduce bias. 3) **Stacking** [9, 18, 6]: It is an ensemble learning technique where multiple models are trained and their outputs are combined using another model, called a meta-model. The meta-model is trained on the outputs of the base models and can learn to combine their strengths. 4) **Gradient Boosting** [13, 7, 20]: It is a technique that combines boosting with gradient descent. Each weak learner is trained to minimize the residual error of the previous learner. Gradient Boosting is widely used in computer vision and can achieve state-of-the-art performance on many computer vision tasks.

Shapley learning A lot of efforts have been invested in exploring the Shapley within different stages of the machine learning process, including trustworthy AI [36, 16, 26], feature selection [45, 39, 23], data valuation [10, 19], and ensemble pruning [35]. In the context of trustworthy AI [36, 16], Shapley value is used for understanding the black-box deep learning model by estimating the importance of each data input in achieving the model output. In the con-

text of feature selection [45, 39, 23], Shapley value might be used to distinguish among the relevant features the non relevant ones in the training process. It can also be used in the data valuation [10, 19], where the goal is to predict the goodness of fit achieved by a model on the test data. Only one work explored Shapley for ensemble pruning [35], where the target is to determine the importance of the models in the ensemble classifier. Nevertheless, the former work suffers from several limitations: 1. It considers binary coalition based on the number of corrected samples on each classifier.2. It used Monte Carlo approximation [37] which does not compute the contributions of all subsets of models.3. It does not go for end-to-end framework where they only studied the contribution of the models and they did not explain how this information may be beneficial in both the training and in the inference stages. 4. The solution is only limited for classification task, and for graph-based representation.

Discussion Ensemble learning averages the output of several learning models trained independently. Even these methods outperform single learning models, they have several shortcomings: 1. The memory and time complexity linearly increase with the ensemble size (increasing in the number of models trained). 2. Models of poor quality greatly influence the best models in the ensemble. In addition, the Shapley value has been largely studied for problems related to various applications including explanation, feature selection, and data valuation. To our knowledge, only one work that explore ensemble pruning [35], however, it only provides binary coalition with Monte Carlo approximation. It did not explain how the determination of the model importance can be used for both the training and the inference stages. It also designed only for solving classification problem for graph-based data representation. Our contribution with SDL is aligned with ensemble pruning and develop a robust consensus method for general-purpose vision systems. Moreover, SDL is generic and might be applied to other data representation including time series, texts, and graphs.

3. SDL: Shapley Deep Learning

3.1. Principle

First, we will discuss the major elements of the SDL approach for general-purpose vision systems. The developed SDL-based consensus method makes use of deep learning and Shapley value as illustrated in Figure 1. Several deep learning models are trained in the learning phase and the Shapley value is used to select the best model(s) that will be executed in the inference phase and this for each testing image. The information returned by the Shapley value is utilized to determine which model(s) are appropriate during the inference phase. This section contains a detailed description of the SDL components.

3.2. Training

We consider the set of l images used in the training $I = \{I_1, I_2...I_l\}$. The training is performed using the set of n models $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, ...\mathcal{M}_n\}$. Each image I_i is injected to each model $\mathcal{M}_j$ for the training. The loss value v_{ij} is determined by computing error between the output of the model $\mathcal{M}_j$, and the ground truth associated to the image I_i. It will be computed using the loss functions according to the problem at hand. For instance, Binary Cross-Entropy Loss might be used for classification problem, as follows:

$$v_{ij}(y_i, y_{ij}^*) = -y \times log(y_{ij}^*) - (1 - y_i) \times log(1 - y_{ij}^*) \quad (1)$$

y_i is the ground truth value of the image I_i, and y_{ij}^* is the predicted value of I_i by the model $\mathcal{M}_j$. Afterwards, the average loss $\mathcal{L}_i$ of all images in I is determined for each model $\mathcal{M}_i$ as,

$$\mathcal{L}_i(\mathcal{M}_i, I) = \frac{\sum_{I_j \in I} v_{ij}}{l} \quad (2)$$

Definition 1 (Model Output) *We define the set of outputs of the model $\mathcal{M}_j$ by the union of all outputs of this model when training the set of images in I, and we write:*

$$\mathcal{Y}_j^* = \{\bigcup_{I_i \in I} y_{ij}^*\} \quad (3)$$

Definition 2 (Model Cost) *Consider t_j and m_j be the runtime, and the memory costs of the model $\mathcal{M}_j$, respectively. We define the cost of the model $\mathcal{M}_j$ by the aggregation of runtime, and the memory costs, and we write:*

$$\mathcal{C}_j = \alpha_1 \times Normalize(t_j) + \alpha_2 \times Normalize(m_j) \quad (4)$$

α_1, and α_2 are the user parameters chosen in the range [0-1]. Normalize is a function min-max normalization [41].

hyperparameter optimization For the hyperparameter optimization of the n models, we adopt the recent greedy search algorithm (GHO) [32]. In order to converge to the local optimal solution with the hope that this decision will result in a global optimal one, the GHO algorithm optimizes each hyperparameter while holding the others constant. Up until all of the hyperparameters are optimized, the local solution for each one is optimized iteratively. Therefore, the greedy algorithm reduces the exponential computational cost of the hyperparameter optimization.

At the end of this step, the set of model players, noted, $\mathcal{MP}$ is created, such as: $\mathcal{MP} = \{(\mathcal{M}_1, \mathcal{L}_1, \mathcal{Y}_1^*, \mathcal{C}_1), (\mathcal{M}_2, \mathcal{L}_2, \mathcal{Y}_2^*, \mathcal{C}_2)...(\mathcal{M}_n, \mathcal{L}_n, \mathcal{Y}_n^*, \mathcal{C}_n)\}$

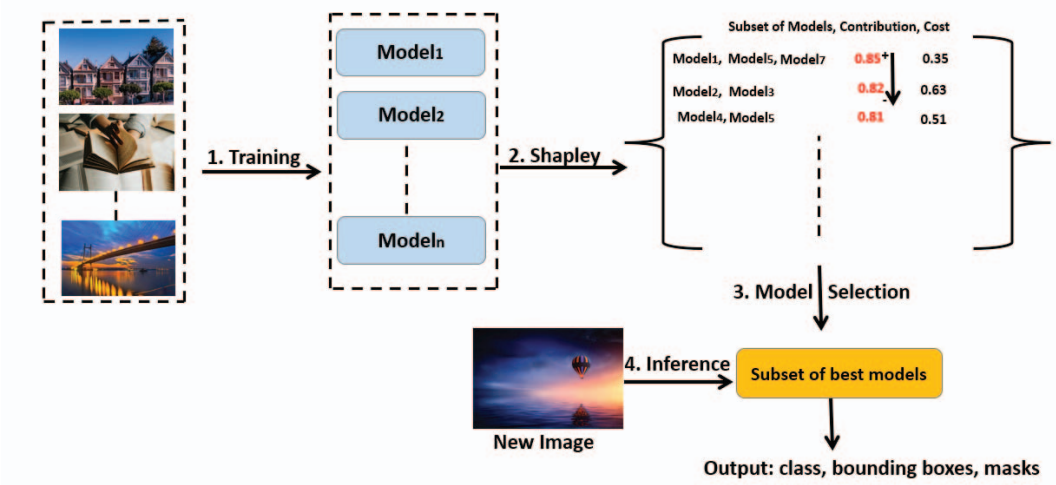

Figure 1: SDL principle: The deep learning models are first trained. The Shapley value is then computed for each subset of models. The best models are finally selected for the inference of new images based on the Shapley value, and their costs.

3.3. Shapley Learning

This step aims to determine the contribution of each model in $\mathcal{MP}$ in the learning process. We draw inspiration from the so-called solution *concepts* or *values* from cooperative game theory. They measure the importance of each player in, or contribution to, a coalitional game. While there are numerous ways to evaluate each player's significance, some solution concepts are seen as being more basic than others because of the axiom systems that define them specifically. The *Shapley value* is a significant game-theoretic solution notion that has sparked a great deal of interest in the area of deep learning [1]. In the following, we will show how we can adapt the Shapley value to calculate the importance of the models in $\mathcal{M}$ in the learning process.

Definition 3 (Shapely for Models) *Let us denote by $\langle c, \mathcal{MP}\rangle$, a coalition game where $c : 2^{\mathcal{MP}} \rightarrow \mathbb{R}$, is the set function that assigns utility to each coalition of the subset of players in $\mathcal{MP}$. Then, the Shapley value of the subset of model players $p \in \mathcal{MP}$ is defined as follows:*

$$\phi_p = \frac{\sum_{F \subseteq \mathcal{MP} \setminus \{p\}} \binom{n-1}{|F|}^{-1} (c(F \cup \{p\}) - c(F))}{n} \quad (5)$$

We also need to define the function c, which expands the training to all subsets of models $F \subseteq \mathcal{MP}$, in order to standardize this formula in our setting. Thus, the function c allows us to drop models in $\mathcal{MP} \setminus F$ based on loss values $\{\mathcal{L}_i\}$ of the models $\{\mathcal{M}_i\}$.

Definition 4 (Loss Coalition) *Consider a subset $p \in \mathcal{MP}$, we define the loss coalition value of p by the average loss values of all models in p as,*

$$c(p) = \frac{\sum_{\mathcal{MP}_i \in p}(1 - \mathcal{L}_i)}{|p|} \quad (6)$$

Normalization is performed to ensure that the loss values should be between 0 and 1.

Definition 5 (Data Coalition) *Consider a subset $p \in \mathcal{MP}$, we define the data coalition value of p by the maximum loss values of all models in p compared to the set $\mathcal{Y}$, composed by the ground-truth of all images in I, as,*

$$c(p) = \frac{\sum_{I_j \in I} max_{\mathcal{MP}_i \in p}\{v_{ij}(y_{ij} \in \mathcal{Y}, y^*_{ij} \in \mathcal{Y}^*_i)\}}{l} \quad (7)$$

Definition 6 (Joint Model Cost) *Consider a subset $p \in \mathcal{MP}$, we define the joint cost of the p by the average cost of all models in p, and we write:*

$$\mathcal{J}_p = \sum_{\mathcal{MP}_j \in p} \mathcal{C}_j \quad (8)$$

To determine the coalition of all subsets in $\mathcal{MP}$, a shapley value method is developed. The process starts by exploring the tree-search of the player models, noted $\mathcal{T}$. As we are not interested to a singleton subset, $2^{|\mathcal{MP}|} - |\mathcal{MP}|$ nodes need to be generated and explored, where at every pass, the coalition of the node, and the joint model cost are calculated. This process is repeated for all subsets in $\mathcal{MP}$. Computing Shapley value is high time consuming, in particular when the number of models increased and with using the data coalition function described in EQ. 7.

The Shapley value is not the sole theory-game-based approach that has been promoted for figuring out how much

each player contributes to the game. The *Banzhaf value* is the most well-studied alternative for Shapley value coming from the coalitional game theory. This value accumulates the contributions of individuals differently, such as:

$$\beta_p = \frac{1}{2^{n-1}} \sum_{F \subseteq \mathcal{MP} \setminus \{p\}} \left(c(F \cup \{p\}) - c(F) \right). \quad (9)$$

The Shapley value is a weighted average of participants' marginal coalition contributions, whereas the Banzhaf value would be a simple average. This will reduce the complexity of the Shapley value computation. We will explore both formulas (EQ. 5, and EQ. 9) in computing the contribution of each subsets of models in $\mathcal{MP}$.

Once all subsets are generated, the subsets of models are ranked according to their contribution value in a descending order. At the end of this step, the set $\mathcal{P}$ of the subsets $P_i \subseteq \mathcal{MP}$ are created with their contributions, and joint costs, as illustrated in:

$$\mathcal{P} = \begin{pmatrix} p_1 & \phi_{P_1} & \mathcal{J}_{P_1} \\ p_2 & \phi_{P_2} & \mathcal{J}_{P_2} \\ . & & \\ . & & \\ . & & \\ p_{2^{|\mathcal{MP}|}-\mathcal{MP}} & \phi_{P_{2^{|\mathcal{MP}|}-\mathcal{MP}}} & \mathcal{J}_{P_{2^{|\mathcal{MP}|}-\mathcal{MP}}} \end{pmatrix}$$

In case the Banzhaf function is considered, ϕ_p is replaced by β_p

Model Selection This step aims to select the best models that will be used for the inference. We consider a cost threshold μ which represent the maximal budget cost that should not exceed in the learning process. It is based on both the runtime, and the memory costs, and it is normalized using *Normalize* function. The process starts by exploring the sorted set $\mathcal{P}$. We compare the budget constraint μ with the joint cost of P_1, if P_1 satisfied the budget constraint, then P_1 is returned, and we terminate the search process. Otherwise, the process is repeated for P_2 and so on until we found a subset that satisfies the budget constraint μ. At the end of this step, the set of best models p_{best} is returned.

Inference The selected best models in p_{best} is used for the inference step. Let us consider the new image I', $y'_{best}(i)$ is the inference output of the model $\mathcal{M}^i_{best}$ on the new image I'. We will use two strategies to explore the selected best models for inference the final output y'_{best}:

1. **Average voting**: In the average voting, the output of the best models are averaged without any weights as follows:

$$y'_{best} = \frac{\sum_{\mathcal{M}^i_{best} \in p_{best}} y'_{best}(i)}{|p_{best}|} \quad (10)$$

2. **Weighted voting**: In the weight voting, the output of the best models are averaged by considering the importance of each model in the set p_{best} as follows:

$$y'_{best} = \frac{\sum_{\mathcal{M}^i_{best} \in p_{best}} w^i_{best} \times y'_{best}(i)}{\sum_{\mathcal{M}^i_{best} \in p_{best}} w^i_{best}} \quad (11)$$

where w^i_{best} represents the weight of the model $\mathcal{M}^i_{best}$. It is calculated by the ratio of the importance of the model $\mathcal{M}^i_{best}$ in the set p_{best}, and it is given as,

$$w^i_{best} = \frac{\mathcal{L}^i_{best}}{\phi_{best}} \quad (12)$$

In case we consider the Banzhaf function, ϕ_{best} is replaced by β_{best}.

Algorithm 1 SDL($\mathcal{M}$, I, $\mathcal{Y}$, μ)

1: $\mathcal{MP} \leftarrow \emptyset$
2: **for** each $\mathcal{M}_i \in \mathcal{M}$ **do**
3: $\quad \mathcal{Y}^*_i \leftarrow \emptyset$
4: $\quad$ **for** each $I_j \in I$ **do**
5: $\quad\quad y^*_{ji} \leftarrow ForwardPropagation(I_j)$
6: $\quad\quad \mathcal{Y}^*_i \leftarrow \mathcal{Y}^*_i \cup \{y^*_{ji}\}$
7: $\quad$ **end for**
8: $\quad \mathcal{W}_i \leftarrow GHO(BackPropagation(\mathcal{Y}^*_i))$
9: $\quad \mathcal{L}_i \leftarrow ComputeLoss(\mathcal{M}_i, I, \mathcal{Y}, \mathcal{Y}^*_i)$
10: $\quad \mathcal{C}_i \leftarrow DetermineCost(\mathcal{M}_i))$
11: $\quad \mathcal{MP} \leftarrow \mathcal{MP} \cup \{(\mathcal{M}_i, \mathcal{L}_i, \mathcal{Y}^*_i, \mathcal{C}_i)\}$
12: **end for**
13: $P \leftarrow \emptyset$
14: **for** each $p \subseteq \mathcal{MP}$ **do**
15: $\quad$ **if** Shapley **then**
16: $\quad\quad P \leftarrow P \cup \{p, \phi_p, \mathcal{J}_p\}$
17: $\quad\quad P \leftarrow Sort(P, \phi)$
18: $\quad$ **else**
19: $\quad\quad P \leftarrow P \cup \{p, \beta_p, \mathcal{J}_p\}$
20: $\quad\quad P \leftarrow Sort(P, \beta)$
21: $\quad$ **end if**
22: **end for**
23: **for** each $P_i \in P$ **do**
24: $\quad$ **if** $\mathcal{J}_{P_i} \leq \mu$ **then**
25: $\quad\quad p_{best} \leftarrow P_i$
26: $\quad\quad$ **break**:
27: $\quad$ **end if**
28: **end for**
29: **return** p_{best}

Algorithm Algorithm 1 presents the formal description of the SDL steps. It takes as input the set of the models $\mathcal{M}$, the set of images I, with their ground-truth $\mathcal{Y}$, and the maximum budget constraint μ. The process starts by training the models in $\mathcal{M}$, from line 1 to line 12. The output of this step is the set of model players $\mathcal{MP}$ with its relevant information of the lost, the cost, and the outputs of the models. The Shapley value, and the joint cost are determined for each subset in $\mathcal{MP}$ from line 13 to line 22. The output of this step will be the sorted set P of all possible subsets of the models players with their Shapley value, and the joint cost. The process ends by selecting the best models in P that maximize the Shapley value and satisfies the maximal

budget constraint μ (from line 23 to line 28). The SDL algorithm will return the set of the best models p_{best} that will be used in the inference.

4. Numerical Results

To evaluate the SDL approach, intensive simulation have been carried out using well-known benchmarks, and compared with recent deep learning solutions in solving computer vision tasks.

4.1. Setting Details

We will first go through the details of our experiment in this section. Then, we will compare our results to those of baseline models. Since ImageNet, COCO, and MNIST have been ones of the most thoroughly benchmarked datasets in computer vision and since advancements on ImageNet, COCO, and MNIST transfer to other datasets [33, 17], we undertake experiments on the ImageNet 2012, and MNIST challenge classification task, and on the COCO challenge object detection task. Further to the GHO algorithm employed in SDL for hyperparameter optimization, and inspired by the work of Xie et al. [44], we optimize again the batch size, and the number of epochs. We utilize a batch size of 2048 by default for labeled images, and we decrease the batch size when the model cannot fit in the memory. We discover that employing 512, 1024, or 2048-batch sizes results in the same speed. The batch size for labeled images is used to calculate the number of training epochs and the learning rate. With a dropout rate of 0.5, we apply dropout to the last layer of the models in SDL and the baseline models. α_1, and α_2 are set equally to 0.5 each. The maximal budget cost μ is set to the average model cost of all models in SDL setting.

4.2. Shapley Vs. Banzhaf

This first experiment aims to understand the impact of using Banzhaf heuristic compared to the Shapley heuristic. Starting by comparing the running times of the Shapley and Banzhaf algorithms. Figure 2 depicts the Shapley and Banzhaf runtimes on generated synthetic data. It is built on two identically shaped subtrees, both of which are full binary trees with a depth of 15. By varying the depth size from 1 to 15, it is clear that Banzhaf is faster than Shapley, which can result in significant time savings for larger trees. Both solutions converge for small trees, but for large trees, there is a significant difference in runtime between the two methods. This is explained by Banzhaf considering the marginal contribution of each model averaged across all possible coalitions that do not include that model. Shapley, on the other hand, considers each model's marginal contribution averaged across all permutations. To compare the quality of the returned outputs of Shapley and Banzhaf, we used the average Cayley distance [12] between models orderings derived from Shapley and Banzhaf values. Figure 2 presents the average Cayley distance between both algorithms (Shapley and Banzhaf) while varying the depth size from 1 to 15. The results indicate high convergence between these two models, whatever the depth size. From these promising results, we will use the Banzhaf heuristic in the remaining experiments. The second experiment of this initial tests is to analyze the cost of using the loss coalition, and the data coalition functions on the Banzhaf heuristic. We used three different models as players namely VGG16 [38], DenseNet [11], and Inception [2] using MNIST data. We varied the percentage of MNIST from 20% to 100%, and we compute the runtime of Banzhaf using loss coalition, and data coalition. The results are reported in Figure 2. The results indicates the stability of loss coalition function compared to the data coalition function. Indeed the runtime of the data coalition is increased by increasing in the training data size. This result is explained by the fact the data coalition function is dependent to the data, where the loss coalition is only dependent to the loss values of the models. Even there is a high gap between the runtime of the loss coalition, and the data coalition, these functions are only used in the training process, and might be executed offline. We will show in further analysis the quality of loss coalition, and data coalition functions for both classification and object detection tasks.

4.3. SDL Vs. Advanced Deep Learning Solutions

This experiment analyzes the performance of the SDL on two case studies (image classification, and object detection), and with the following advanced deep learning solutions:

1. **Classification**: We use two algorithms for comparison regarding the classification task, namely Revised RESNET [5], MViTv2 [24], and LL-R [22]. We use the combination of the three models as model players for SDL. We use the weighted voting mechanism in the inference.

2. Object Detection: We use three algorithms for comparison regarding the object detection task, namely MViTv2 [24], FSOD [14], and Improved Yolov5 [31]. We use the combination of the three models as model players for SDL. We use the weighted voting mechanism in the inference.

Using the previously described ImageNet, these experiments compare the SDL's accuracy against SOTA image classification methods (Revised RESNET, MViTv2, and LL-R). Figure 3 demonstrates that SDL outperforms the two baseline algorithms in terms of classification rate and it is very competitive in terms of inference runtime when the percentage of the number of images used as input is varied from 20% to 100%. Thus, the classification rate of the SDL is 95% whereas the baseline methods go below 90% when the entire ImageNet is processed in the train-

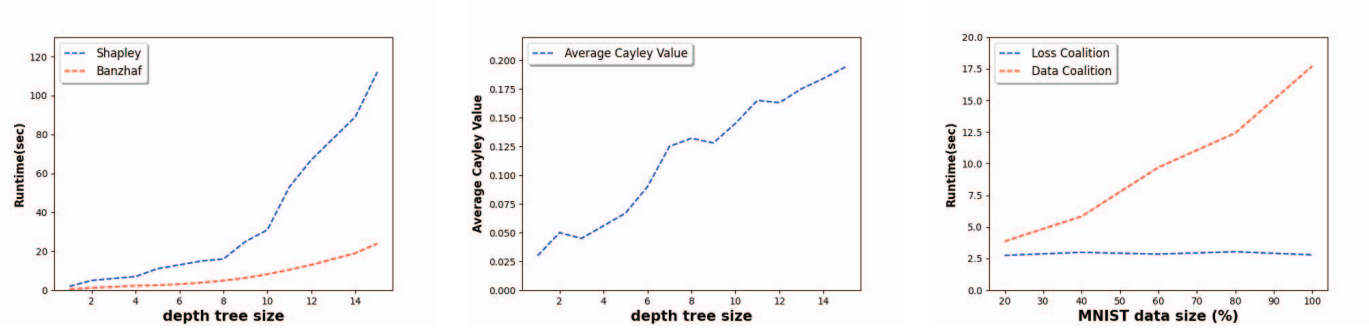

Figure 2: Performance of the Shapely value compared to the Banzhaf.

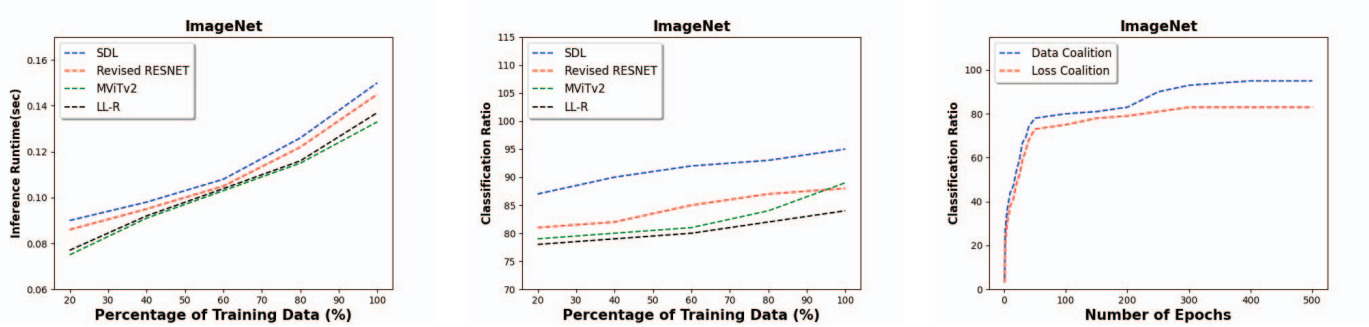

Figure 3: Classification rate and Runtime of the proposed solution and the SOTA models for different training samples of the ImageNet, and with different number of epochs.

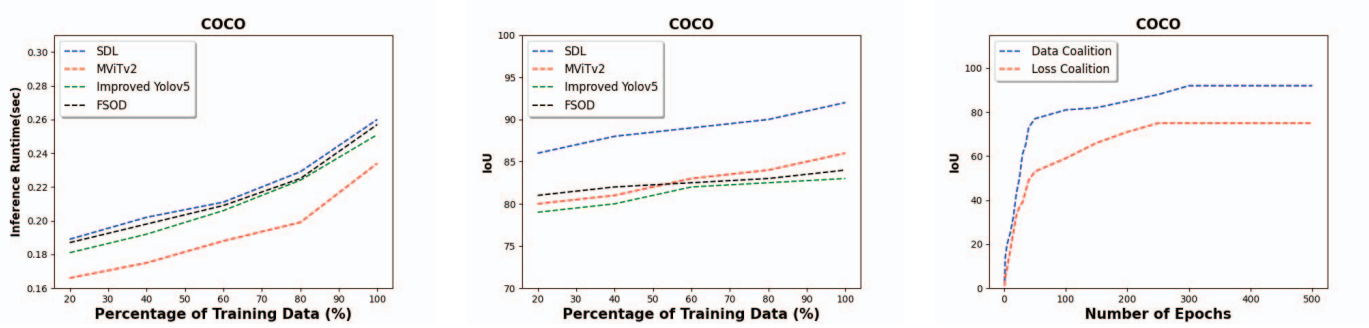

Figure 4: Performance of the proposed solution and the SOTA models for object detection use case using COCO dataset, and with different number of epochs.

ing phase. These results are obtained thanks to the selective strategy based on Shapley value, and the loss coalition function, which achieve in finding the best models executed in the inference stage. Using the previously described COCO dataset, the next experiments compare the SDL's accuracy against SOTA object detection methods (MViTv2, Yolov5, and FSOD). Figure 4 demonstrates that SDL achieved a great performance compared to the two baseline algorithms in terms of IoU (Intersection over Union) and it is very competitive in terms of inference runtime when the percentage of the number of images used as input is varied from 20% to 100%. Thus, the IoU of the SDL is 92% whereas the baseline methods go below 86% when the entire COCO dataset is processed in the training phase. These outcomes were again made possible by the selective strategy based on the shapley value, which looked through all the models and determines the contribution of each model to identify the most appropriate models for the inference stage.

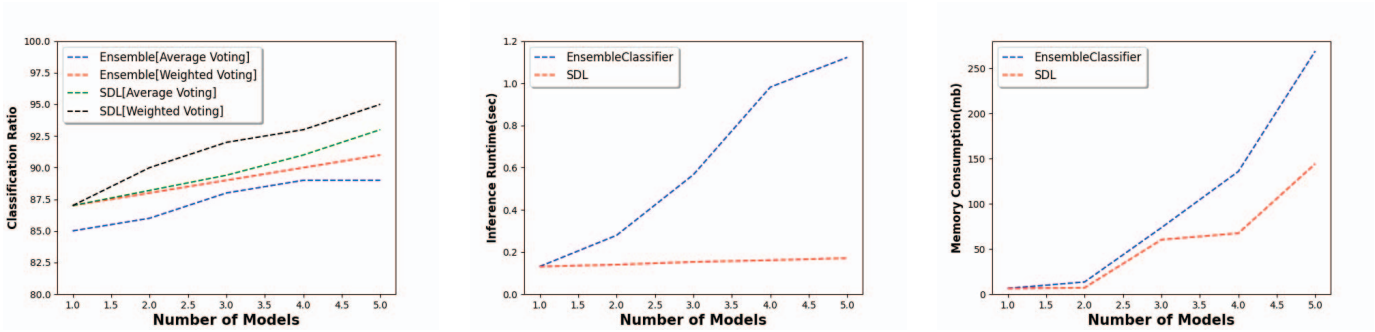

Figure 5: Performance of SDL compared to ensemble learning based solutions.

4.4. Qualitative Results for Coalition Functions

The next experiments aims to evaluate the coalition functions developed in SDL. We varied the number of epochs from 1 to 500 while fixing the remaining optimal parameters. The results highlighted in Figure 3, and Figure 4 show that while increasing the number of epochs, the classification rate, and the IoU increased. For instance, when the number of epochs is set to 5, the classification rate of SDL does not exceed 36%. However, when the number of epochs is set to 500, the classification rate of SDL reaches 95%. In addition, the data coalition gives better results compared to the loss coalition, whatever the number of epochs, and the scenario used in the experiment. These results explain the powerful function of the data coalition compared to the loss coalition. Indeed, the loss coalition considers directly the loss functions of the models in computing the coalition, where the data coalition takes into account the model output for deriving the coalition value of the models.

4.5. SDL Vs. Ensemble Learning

The last experiments aims to compare SDL with the ensemble learning strategy using the ImageNet data. Further to VGG16, DenseNet, and Inception, two other models (RESNET [21], and AlexNet [27]) are also used in both the ensemble classifier, and SDL. To make a fair comparison, the weights of the ensemble classifier with weighted average strategy are the same weights as SDL with the weighted average strategy. The order of the models are as follows {Inception, DenseNet, AlexNet, RESNET, and VGG16}. By varying the number of models from 1 to 5, Figure 5 shows that the classification ratio of SDL is better than the ensemble classifier, whatever the scenario used. Indeed, the classification rate of SDL achieved 95%, where the ensemble classifier does not exceed 91%. These promising results are achieved thanks to the efficient coalition function, and the weighted voting strategy to first compute the contribution of each subset of models, and then explore the best models returned by the banzhaf value heuristic. In addition, we observe high gap in terms of inference runtime, and memory consumption between SDL, and ensemble classifier. This can be explained by the fact that SDL explores a few subsets of models in the inference instead of the ensemble classifier where it needs to make the inference from all models.

5. Conclusion

This work addresses the challenges related to establishing general-purpose and flexible vision systems using the currently existing deep learning models, and propose a novel foundation called SDL for establishing a task-agnostic consensus modelling. For each set of visual data, SDL makes use of many deep learning models for training. Following that, the Shapley value is then determined to compute the contribution of each subset of models in the training. The model selection is finally performed based on the Shapley value and the joint model cost. Optimization of the Shapley computation is also carried out by investigating the Banzhaf function. ImageNet, COCO, and MNIST benchmarks were used to evaluate the designed SDL approach on several tasks. The outcomes presented validated the SDL's higher accuracy and strong inference runtime competitiveness compared to the baseline methods for both classification, and object detection tasks. Since the runtime of the SDL is critical, in particular when the number of models became high, and for real-time processing based applications, we plan to improve the model exploration by investigating other heuristics than Shapley, and Banzhaf values. Investigating SDL for other computer vision tasks and case studies is also on our future agenda.

Acknowledgement This work is co-funded by the Research Council of Norway under the project entitled "Next Generation 3D Machine Vision with Embedded Visual Computing" with grant number 325748.

References

[1] Lucas Agussurja, Xinyi Xu, and Bryan Kian Hsiang Low. On the convergence of the shapley value in parametric bayesian learning games. In *International Conference on Machine Learning*, pages 180–196. PMLR, 2022. 4

[2] Md Zahangir Alom, Mahmudul Hasan, Chris Yakopcic, and Tarek M Taha. Inception recurrent convolutional neural network for object recognition. *arXiv preprint arXiv:1704.07709*, 2017. 6

[3] Khaled Bayoudh, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa. A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets. *The Visual Computer*, 38(8):2939–2970, 2022. 1

[4] Asma Belhadi, Youcef Djenouri, Vicente Garcia Diaz, Essam H Houssein, and Jerry Chun-Wei Lin. Hybrid intelligent framework for automated medical learning. *Expert Systems*, 39(6):e12737, 2022. 1

[5] Irwan Bello, William Fedus, Xianzhi Du, Ekin Dogus Cubuk, Aravind Srinivas, Tsung-Yi Lin, Jonathon Shlens, and Barret Zoph. Revisiting resnets: Improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 34:22614–22627, 2021. 6

[6] Shouzhi Chen, Yanhong Liao, Jianping Zhao, Yannan Bin, and Chunhou Zheng. Pacvp: Prediction of anti-coronavirus peptides using a stacking learning strategy with effective feature representation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2023. 2

[7] Tusar Kanti Dash, Chinmay Chakraborty, Satyajit Mahapatra, and Ganapati Panda. Gradient boosting machine and efficient combination of features for speech-based detection of covid-19. *IEEE Journal of Biomedical and Health Informatics*, 26(11):5364–5371, 2022. 2

[8] Youcef Djenouri, Asma Belhadi, Djamel Djenouri, Gautam Srivastava, and Jerry Chun-Wei Lin. Intelligent deep fusion network for anomaly identification in maritime transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 2022. 1

[9] Xingning Dong, Tian Gan, Xuemeng Song, Jianlong Wu, Yuan Cheng, and Liqiang Nie. Stacked hybrid-attention and group collaborative learning for unbiased scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19427–19436, 2022. 2

[10] Zhenan Fan, Huang Fang, Zirui Zhou, Jian Pei, Michael P Friedlander, Changxin Liu, and Yong Zhang. Improving fairness for data valuation in horizontal federated learning. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 2440–2453. IEEE, 2022. 2, 3

[11] Zhenyu Fang, Jinchang Ren, Stephen Marshall, Huimin Zhao, Song Wang, and Xuelong Li. Topological optimization of the densenet with pretrained-weights inheritance and genetic channel selection. *Pattern Recognition*, 109:107608, 2021. 6

[12] Michael A Fligner and Joseph S Verducci. Distance based ranking models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 48(3):359–369, 1986. 6

[13] Magzhan Gabidolla and Miguel Á Carreira-Perpiñán. Pushing the envelope of gradient boosting forests via globally-optimized oblique trees. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 285–294, 2022. 2

[14] Bin-Bin Gao, Xiaochen Chen, Zhongyi Huang, Congchong Nie, Jun Liu, Jinxiang Lai, Guannan Jiang, Xi Wang, and Chengjie Wang. Decoupling classifier for boosting few-shot object detection and instance segmentation. In *Advances in Neural Information Processing Systems*, 2022. 6

[15] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshik. Mask r-cnn. In *Proceedings of the International Conference on Computer Vision*, pages 2980–2988, 2017. 2

[16] Tom Heskes, Evi Sijben, Ioan Gabriel Bucur, and Tom Claassen. Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. *Advances in neural information processing systems*, 33:4778–4789, 2020. 2

[17] Zhi Hou, Baosheng Yu, Yu Qiao, Xiaojiang Peng, and Dacheng Tao. Affordance transfer learning for human-object interaction detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 495–504, 2021. 6

[18] S Iniyan, Anurag Singh, and Brishti Hazra. Wavelet transformation and vertical stacking based image classification applying machine learning. *Biomedical Signal Processing and Control*, 79:104103, 2023. 2

[19] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1167–1176. PMLR, 2019. 2, 3

[20] Xiaojun Jia, Yong Zhang, Baoyuan Wu, Jue Wang, and Xiaochun Cao. Boosting fast adversarial training with learnable adversarial initialization. *IEEE Transactions on Image Processing*, 31:4417–4430, 2022. 2

[21] Heechul Jung, Min-Kook Choi, Jihun Jung, Jin-Hee Lee, Soon Kwon, and Woo Young Jung. Resnet-based vehicle classification and localization in traffic surveillance systems. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 61–67, 2017. 8

[22] Youngwook Kim, Jae Myung Kim, Zeynep Akata, and Jungwoo Lee. Large loss matters in weakly supervised multilabel classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14156–14165, 2022. 6

[23] Guanghui Li, Jiahua Shen, Chenglong Dai, Jia Wu, and Stefanie I Becker. Shveegc: Eeg clustering with improved cosine similarity-transformed shapley value. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2022. 2, 3

[24] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022. 6

[25] Zongyi Li, Yuxuan Shi, Hefei Ling, Jiazhong Chen, Qian Wang, and Fengfan Zhou. Reliability exploration with self-ensemble learning for domain adaptive person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1527–1535, 2022. 1

[26] Weixin Liang, Girmaw Abebe Tadesse, Daniel Ho, L Fei-Fei, Matei Zaharia, Ce Zhang, and James Zou. Advances, challenges and opportunities in creating data for trustworthy ai. *Nature Machine Intelligence*, 4(8):669–677, 2022. 2

[27] Xianxian Luo, Wenghao Wen, Jingru Wang, Songya Xu, Yingying Gao, and Jianlong Huang. Health classification of meibomian gland images using keratography 5m based on alexnet model. *Computer Methods and Programs in Biomedicine*, 219:106742, 2022. 8

[28] Sparsh Mittal, Srishti Srivastava, and J Phani Jayanth. A survey of deep learning techniques for underwater image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 1

[29] Yujian Mo, Yan Wu, Xinneng Yang, Feilin Liu, and Yujun Liao. Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing*, 493:626–646, 2022. 1

[30] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019. 1

[31] Zhong Qu, Le-yuan Gao, Sheng-ye Wang, Hao-nan Yin, and Tu-ming Yi. An improved yolov5 method for large objects detection with multi-scale feature cross-layer fusion network. *Image and Vision Computing*, 125:104518, 2022. 6

[32] Goutham Rajendran, Bohdan Kivva, Ming Gao, and Bryon Aragam. Structure learning in polynomial time: Greedy algorithms, bregman information, and exponential families. *Advances in Neural Information Processing Systems*, 34:18660–18672, 2021. 3

[33] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019. 6

[34] Vahid Rowghanian. Underwater image restoration with haar wavelet transform and ensemble of triple correction algorithms using bootstrap aggregation and random forests. *Scientific Reports*, 12(1):8952, 2022. 2

[35] Benedek Rozemberczki and Rik Sarkar. The shapley value of classifiers in ensemble games. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1558–1567, 2021. 2, 3

[36] Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International conference on machine learning*, pages 9269–9278. PMLR, 2020. 2

[37] Sofiane Touati, Mohammed Said Radjef, and SAIS Lakhdar. A bayesian monte carlo method for computing the shapley value: Application to weighted voting and bin packing games. *Computers & Operations Research*, 125:105094, 2021. 3

[38] Ivan Vajs, Vanja Ković, Tamara Papić, Andrej M Savić, and Milica M Janković. Dyslexia detection in children using eye tracking data based on vgg16 network. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 1601–1605. IEEE, 2022. 6

[39] Mochitha Vijayan, SS Sridhar, and Dhinakaran Vijayalakshmi. A deep learning regression model for photonic crystal fiber sensor with xai feature selection and analysis. *IEEE Transactions on NanoBioscience*, 2022. 2, 3

[40] Pichao Wang, Xue Wang, Fan Wang, Ming Lin, Shuning Chang, Hao Li, and Rong Jin. Kvt: k-nn attention for boosting vision transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 285–302. Springer, 2022. 2

[41] Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. In *International Conference on Machine Learning*, pages 23631–23644. PMLR, 2022. 3

[42] Thomas Westfechtel, Hao-Wei Yeh, Qier Meng, Yusuke Mukuta, and Tatsuya Harada. Backprop induced feature weighting for adversarial domain adaptation with iterative label distribution alignment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 392–401, 2023. 2

[43] Eyal Winter. The shapley value. *Handbook of game theory with economic applications*, 3:2025–2054, 2002. 2

[44] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020. 6

[45] Dingze Yin, Dan Chen, Yunbo Tang, Heyou Dong, and Xiaoli Li. Adaptive feature selection with shapley and hypothetical testing: Case study of eeg feature engineering. *Information Sciences*, 586:374–390, 2022. 2, 3

[46] Xingjian Zhen, Zihang Meng, Rudrasis Chakraborty, and Vikas Singh. On the versatile uses of partial distance correlation in deep learning. In *Computer Vision–ECCV 2022: 17th European Conference, 2022, Proceedings, Part XXVI*, pages 327–346. Springer, 2022. 1

[47] Wujie Zhou, Yun Zhu, Jingsheng Lei, Rongwang Yang, and Lu Yu. Lsnet: Lightweight spatial boosting network for detecting salient objects in rgb-thermal images. *IEEE Transactions on Image Processing*, 2023. 2

[48] Lei Zhu, Qian Chen, Lujia Jin, Yunfei You, and Yanye Lu. Bagging regional classification activation maps for weakly supervised object localization. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*, pages 176–192. Springer, 2022. 2

[49] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 2023. 1