Combining Early Exit and Selective Prediction for Convolutional Neural Networks

Hasna Bouraoui, Chadlia Jerad, and Jeronimo Castrillon

***Abstract*—The deployment of CNN in real-time and resource-constrained applications poses critical challenges due to their computational demands and the need for reliable decision making. In this paper, we combine adaptive inference via Early Exits (EE) with Selective Prediction (SP) to address these challenges. Early exits allow confident predictions at intermediate layers, while selective prediction introduces uncertainty estimation modules, enabling the system to abstain from low-confidence decisions or continue inference through deeper layers. This combined design lowers the risk of overconfident but erroneous predictions and improves the trade-off between performance and accuracy. As a case study, we implement and evaluate our approach on a real-time traffic sign detection task, processing the input of an RGB camera in the forward direction. In this paper, we demonstrate improved performance compared to baseline models. Compared to SP-only (Selective Prediction) and EE-only (Early Exit) baselines, our hybrid model achieves low inference depth (1.20), leading to reduced computational demand and latency. Despite this efficiency, the model maintains a high prediction accuracy (90.3%) and a low abstention rate (1.6%), ensuring fast and reliable decision making suitable for time-critical embedded applications. This demonstrates an effective trade-off between effective computation and predictive reliability.**

***Index Terms*—Early Exit, Selective Prediction, Trade-offs, Convolutional Neural Networks**

I. Introduction

NOWADAYS, Convolutional Neural Networks (CNNs) achieve state-of-the-art results across various domains. However, their high computational cost and latency present challenges for reactive, time-critical embedded systems. This is especially critical in applications like autonomous driving, real-time medical diagnostics, and edge computing. To deal with these constraints, two strategies gained attention: Selective Prediction (SP) and adaptive inference. SP gives the model the option to not make a prediction when it's unsure, typically when its confidence drops below a given threshold [1]. This is especially useful in safety-critical contexts, where uncertain cases can be deferred to humans, enhancing decision reliability. Adaptive inference [2], on the other hand, takes a different approach. Instead of running every input all the way through the model, it adjusts computation on the fly. A prominent example is early exits (EE) — where the model makes predictions partway through, if it's confident enough. The model then exit early for simpler inputs, cutting down computation during inference while maintaining decent accuracy for straightforward cases. In practice, EE strategies have proven effective at lowering both latency and energy use [3]. This relies on the assumption that the intermediate classifiers can make confident decisions on some inputs. While this holds true in many scenarios, it doesn't always generalize across all tasks or deployment settings. Selective prediction and adaptive inference both have clear benefits, but utilizing them separately has some drawbacks. SP on its own still requires running every input through the entire network, including the abstained ones, resulting in unnecessary computational overhead. While, adaptive inference doesn't allow for abstention. This forces the model to make a prediction in low-confidence scenarios, which is problematic in safety-critical applications. That's why combining both strategies leverages the most of their strengths.

In this paper, we show that by integrating EE mechanisms with an SP strategy in a single architecture, a better balance between efficiency, accuracy, and reliability is achievable for real-time systems where latency and predictability are critical. We present an approach to enable CNN to choose between predicting early, abstaining, or proceeding to deeper layers based on confidence estimates at intermediate exits, thereby enabling time-aware and confidence-driven inference paths that are essential for reactive systems. The key contributions of this work are:

- We propose a unified and configurable hybrid approach that combines *EE* and *SP* to balance runtime efficiency and reliability in CNN.
- We formalize our hybrid approach as a *three-way decision* problem, supporting the runtime computed values: predict, continue, and abstain.
- We evaluate our approach on the German Traffic Sign Recognition Benchmark, achieving up to 3.5× computational speedup with minimal accuracy loss (0.5%), and reduced average exit depth (from 3 to 1.08).

The paper is organized as follows. Section II presents the theoretical background and related work. Section III details the proposed hybrid approach. Section IV illustrates and evaluates it on a traffic sign detection use case. Section V concludes and outlines future directions.

II. Theoretical Background and Related Work

We review in this section existing literature on selective prediction, adaptive inference, and hybrid approaches, providing context for our proposed combined mechanism. SP methods [4] improve the reliability of CNN by abstaining from uncertain predictions. It is often referred to as confidence-based abstention, allows models to retain predictions when confidence is below a predefined threshold. This model is common in applications where safety is very important and

Hasna Bouraoui and Jerronimo Castrillon are with the Chair for Compiler Construction at TU Dresden, Germany (e-mails: hasna.bouraoui@tu-dresden.de;Jeronimo.Castrillon@tu-dresden.de).

Chadlia Jerad is with the University of Manouba, Manouba, Tunisia (e-mail: chadlia.jerad@ensi-uma.tn).

misclassifications can be very expensive, so humans can step in or use a different processing method. In the domain of neural networks, authors in [5] introduced a rejection strategy based on output logits, comparing the highest and second-highest activated logits to guide rejection decision. Authors in [1] proposed a selective classification method to attain a target risk with a defined confidence-rate function, building a baseline for risk-controlled predictions. Recent advances in SP have focused on architectural enhancements [6]. Alternative approaches, such as Deep Gamblers in [7] and Self-Adaptive Training [8] propose an additional category for abstention. Authors in [9] performed a thorough examination of the selection mechanisms of these models and highlighted their weaknesses. Their findings suggest that the good performance of these models is mainly due to the optimization process, leading to a more generalizable model that subsequently enhances the performance of SP.

Adaptive inference [2] reduces CNN computation by adjusting processing based on input complexity. A popular strategy is early exiting (EE), which integrates multiple classifiers at different depths, allowing confident predictions to exit early and avoid unnecessary computation. EE has been applied across domains, including image classification [10]. The distinguishing factor among current methodologies is the selection of confidence metrics, including prediction consistency [11], and output entropy [12].

Combined methods use both confidence and uncertainty to decide what to do next. Authors in [13] came up with ensemble reuse techniques to make better use of resources in early exit networks while still being able to abstain. In contrast to our work, they proposes a reuse strategy by recycling the non predicted samples and do not employ an explicit multi-criteria thresholding. Our approach, however, is based on a three way decision logic, combining EE and SP through calibrated uncertainty thresholds and enabling dynamic routing at each exit point. Numerous studies have suggested deferral-based methodologies, and the choice to predict or defer is dictated by a cost function [14], [15]. To the best of our knowledge, we are the first to apply selective prediction and early exit in a hybrid manner. Our proposed model extends these ideas by implementing a three-way routing mechanism that enables dynamic decisions to exit, continue processing, or abstain, calibrated through multi-criteria optimization.

III. Proposed Approach: Combined EE and SP

In our approach, we enhance a standard CNN by adding intermediate decision blocks, which let the network make early predictions when confidence is high enough. We combine early-exit mechanisms with selective prediction via abstention. The model can choose not to predict if it's unsure. This hybrid setup helps the network balance accuracy with efficiency, adjusting its computational effort based on how complex or uncertain an input is. As shown in Fig. 1 we add exit points at representative intermediate layers, following common practices in the literature. Each exit is extended with a lightweight classifier (exit head) that processes intermediate features. This is identified as the exit depth distribution.

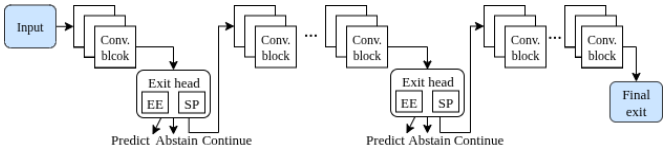

Fig. 1. Combined Approach flow

We define two decision logic, both based on a routing mechanism that evaluates whether the model should make an early prediction, continue processing, or abstain: an entropy-based III-A1 and a confidence-based decisions III-A2. The effective performance of the hybrid decision logic relies on well-calibrated thresholds III-B. We evaluate the computational savings achieved through early exits using the *effective computation* metric. It represents the percentage of the full model's computation used on average, based on the exit depth distribution. This metric quantifies the computational savings achieved by EE. Formally, let L_k be the number of layers up to exit k, and let N_k be the number of samples that exited at exit k. Let L be the total number of layers in the full model. Then, the effective computation is defined as:

$$\text{Effective Computation} = \left(\frac{1}{N} \sum_{k=1}^{K} \frac{L_k}{L} \cdot N_k \right) \times 100\% \quad (1)$$

where K is the number of exits, N is the total number of samples, and $\frac{L_k}{L}$ is the fraction of the model used for exit k.

A. Decision Logic

1) Entropy-Based: Entropy is a scalar measure of uncertainty. It measures the spread of the softmax probability output. Higher entropy means greater uncertainty, while lower entropy means more confident predictions.

The hybrid, entropy-based, model adds several *exit points* to the network architecture, each with a simple classifier. To measure uncertainty, the model computes the *entropy* of the predicted class distribution at each exit. When the entropy is below a predefined threshold τ_{exit}, indicating high confidence, the model predicts early to reduce computational cost. If the entropy exceeds a higher threshold τ_{continue}, suggesting significant uncertainty, the model continues processing through deeper layers to refine its prediction. In cases where the entropy lies between τ_{exit} and τ_{continue}, reflecting moderate uncertainty, the model abstains from making an immediate decision and defers to a more reliable confidence estimator. To quantify the model's uncertainty at each exit point, we compute the entropy of the softmax probability distribution over the class predictions. Formally, let $H_l(x)$ denote the entropy of the softmax output at exit layer l for input x. The model applies this decision logic:

$$\text{Decision}_l(x) = \begin{cases} \text{Predict} & \text{if } H_l(x) < \tau_{\text{exit}} \\ \text{Abstain / Defer} & \text{if } \tau_{\text{exit}} \leq H_l(x) < \tau_{\text{continue}} \\ \text{Continue} & \text{if } H_l(x) \geq \tau_{\text{continue}} \end{cases}$$

Let $\mathbf{p}_l(x) = \text{softmax}(\mathbf{z}_l(x))$ be the class probability distribution produced at exit layer l, where $\mathbf{z}_l(x) \in \mathbb{R}^C$ is the

logit vector for input x and C is the number of classes. Then, the entropy at exit l is defined as:

$$H_l(x) = -\sum_{c=1}^{C} p_{l,c}(x) \log p_{l,c}(x) \tag{2}$$

where $p_{l,c}(x)$ is the probability assigned to class c by the softmax output at layer l.

2) Confidence-Based: Confidence is the maximum probability of the softmax distribution, it shows how sure the model is of its most likely prediction. Low-confidence samples are abstained to avoid unreliable predictions. Inputs with low confidence are passed to further layers, to allow a refinement before the decision. Similarly to the entropy-based hybrid approach, the confidence score is defined as:

$$\text{Conf}_l(x) = \max_{c \in \{1,\ldots,C\}} p_{l,c}(x) \tag{3}$$

The model applies the following decision logic:

$$\text{Decision}_l(x) = \begin{cases} \text{Predict} & \text{if Conf}_l(x) \geq \tau_{\text{predict}} \\ \text{Abstain / Defer} & \text{if Conf}_l(x) \leq \tau_{\text{abstain}} \\ \text{Continue} & \text{otherwise} \end{cases}$$

We note that entropy and confidence measure uncertainty in opposite ways. Entropy increases with uncertainty, while confidence decreases. The decision logic follows a monotonic trend with uncertainty: as uncertainty increases, the model shifts from prediction to abstention, then to deeper inference.

B. Threshold Calibration

Effective performance of the hybrid decision logic relies on well calibrated thresholds for early prediction, abstention, and continued inference. This builds up to identify values for (τ_{continue}) and (τ_{exit}) in entropy-based approach, (τ_{predict}) and (τ_{abstain}) in confidence-based approach. These thresholds regulate the model's trade-off between performance, accuracy and abstention rate. Threshold values are usually tuned for each specific use case or model. They can be determined through calibration procedure. In the case of CNNs, We used a grid search over validation data to approximate suitable thresholds for the decision logic. Our process for calibration includes iteratively testing different entropy values as thresholds to make decisions within a model, and then evaluating on a validation set, how these different thresholds impact the model's performance. We also opted for balancing exit weighting. We selected thresholds that balance the frequency of early exits with the overall accuracy and abstention rate, avoiding premature or overly delayed decisions. Finally, to ensure consistent decision-making across layers, we jointly calibrated the early exit threshold τ_{exit} and the continue threshold τ_{continue}. A grid search on a validation set optimized the trade-off between computational savings, abstention rate, and overall accuracy. Although brute-force, grid search remains popular in early exit and selective prediction research for its simplicity and reproducibility.

IV. Evaluation on a Traffic Sign Detection System

A. Use Case Descripton and Experimental Setup

To evaluate our approach, we use a `MobileNetV3Small`-based CNN for traffic sign classification, suitable for edge deployment. The model has 144 operation-level layers (e.g., convolutions, batch norms), grouped into 13 inverted residual blocks and includes four classifier stages: three internal exits and one final classifier. We placed the exits after block 2, 5, and 11. This corresponds roughly to 22%, 49%, and 94% of total network depth. This exit placement is balanced for easy examples (block 2), efficient mid-path predictions (block 5), and a final semantic content (block 11). We implemented the use case in PyTorch, supporting configurable routing metrics (entropy or confidence), which is also fully compatible with post-training threshold calibration.

We perform extensive experiments to quantify the performance of six model variants across accuracy, abstention, average exit depth, and computational efficiency (Baseline, EE-only entropy based, EE-only confidence based, SP-only, Hybrid model entropy based and confidence based). We evaluated the model for the German Traffic Sign Recognition Benchmark (GTSRB), a standard dataset comprising 43 traffic sign categories with varying real-world visual conditions. Images are resized to 32×32 RGB format and processed through augmentation, normalization, and training-validation splitting. We performed the model training on a system equipped with an NVIDIA RTX 3090 GPU with 24 GB of memory, while the evaluation is done on a workstation with 8 CPU cores.

B. Evaluation

1) Evaluation Metrics: Our goal is to compare each model variant in terms of predictive performance, computational efficiency, and reliability under uncertainty. *Effective Accuracy* measures the correctly classified samples among all samples in the dataset, including the abstained ones. This metric penalizes abstention and provides a comprehensive measure of overall system performance. While *Accuracy on predicted* refers to the number of samples that were correctly classified out of all the predicted samples, highlighting the model's classification performance on confident decisions. *Abstention rate* denotes the proportion of samples for which the model abstains due to uncertainty, reflecting the system's tendency to defer decisions when unsure. Lastly, *Average Exit Depth* captures the mean exit point across all predicted samples. We compute it using normalized block indices, where each exit is associated with the number of computational blocks traversed up to that point. It serves as an indicator of the computational efficiency of the early-exit mechanism, where lower values suggest that more samples are exiting earlier, thereby reducing processing cost.

2) Results Interpretation: For the evaluation, we use the variant without EE or SP as baseline. Table I reports the main performance metrics across all six model variants. The baseline achieves an accuracy of (88.78%). In contrast, the EE Only (Entropy) model attains 89.40% accuracy with an average exit depth of only 1.06, demonstrating the effectiveness of entropy-based early inference. The Hybrid (Entropy) model achieves good prediction reliability (90.32%) while abstaining

TABLE I
PERFORMANCE COMPARISON OF MODEL VARIANTS

Model	Eff. Acc.	Pred. Acc.	Abst.%	Avg. Exit Depth
Baseline	88.78	88.78	0	1
EE (Conf.)	88.83	88.83	0	0.29
EE (Entropy)	89.40	89.40	0	0.25
SP Only	88.12	89.71	1.77	1
Hybrid (Entropy)	88.87	90.32	1.61	0.28
Hybrid (Conf.)	87.09	89.05	2.21	0.25

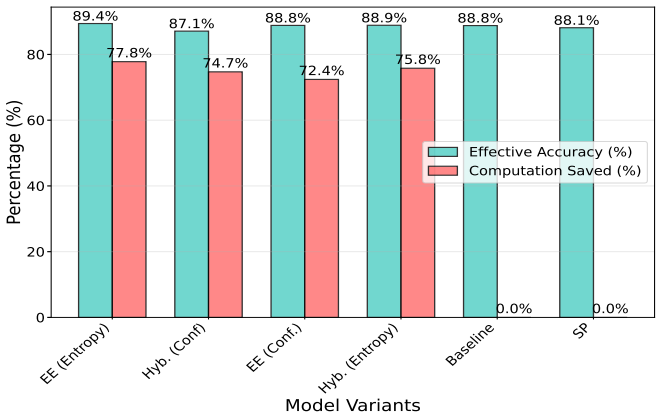

Fig. 2. Accuracy vs Computational Savings Comparison.

on only 1.6% of uncertain samples, demonstrating a well-calibrated balance between accuracy and safety. Meanwhile, the Hybrid (Confidence) model offers greater inference efficiency (avg. exit depth: 1.08) with minimal loss in accuracy, making both hybrid approaches outperforms the traditional baselines for adaptive and reliable neural inference.

Fig. 2 illustrates the trade-off between effective accuracy and computational savings across model variants. Both hybrid models achieve up to 75.8% in computation savings, equivalent to a 3.5× theoretical speedup compared to the full-depth baseline, while maintaining competitive accuracy levels (87–89%). In addition, Fig. 3 highlights the trade-off between effective computation and predicted accuracy of different model variants. The hybrid models (entropy-based) achieve higher predictive accuracy than both EE only and SP only baselines, while using only about 24–25% of its computation. In contrast, the baseline and SP-only models achieve similar accuracy but require full-depth inference, highlighting the hybrid models' ability to trade-off between computational efficiency and prediction trustworthiness. Compared to the four other variants, our hybrid models offer the best balance of accuracy, computational savings, and reliability.

V. CONCLUSION AND FUTURE WORK

Our approach enables CNN systems to trade-off early prediction, abstention, or deeper inference based on confidence estimates. We showed, through the German Traffic Sign Recognition Benchmark, that combining EE and SP reduces computation while maintaining high accuracy and predictive reliability. Future work will investigate tailoring thresholds to exit depths based on input complexity. We will also explore more complex benchmarks to further validate the generality

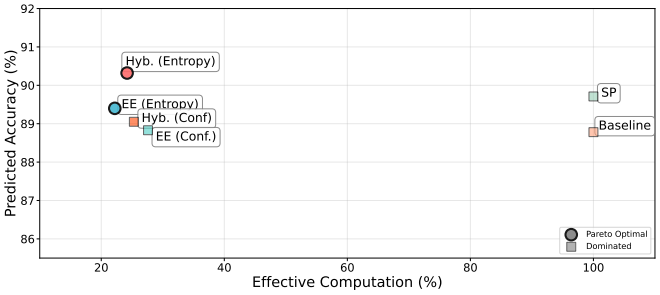

Fig. 3. Effective Computation vs Accuracy Trade-off.

of our hybrid approach. Another direction is to examine alternative exit point placements.

ACKNOWLEDGMENT

This work was funded in part by the EU Horizon Europe Programme under grant agreement No 101135183 (MYRTUS). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

REFERENCES

[1] Y. Geifman and R. El-Yaniv, "Selective classification for deep neural networks," *Advances in neural information processing systems*, vol. 30, 2017.
[2] H. Rahmath P, V. Srivastava, K. Chaurasia, R. G. Pacheco, and R. S. Couto, "Early-exit deep neural network-a comprehensive survey," *ACM Computing Surveys*, vol. 57, no. 3, pp. 1–37, 2024.
[3] X. Li, C. Lou, Y. Chen, Z. Zhu, Y. Shen, Y. Ma, and A. Zou, "Predictive exit: Prediction of fine-grained early exits for computation-and energy-efficient inference," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 7, 2023, pp. 8657–8665.
[4] W. Ye, D. Chen, and I. Ramazanli, "Learning algorithm in two-stage selective prediction," in *2022 Asia Conference on Algorithms, Computing and Machine Learning*. IEEE, 2022.
[5] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," in *NeurIPS*, 1989, pp. 396–404.
[6] C. Cortes, G. DeSalvo, and M. Mohri, "Theory and algorithms for learning with rejection in binary classification," *Annals of Mathematics and Artificial Intelligence*, vol. 92, no. 2, pp. 277–315, 2024.
[7] Y. Liu, , J. Zhang, , and Y. Liu, "Integrating abstention into training: Deep gamblers for reliability," in *NeurIPS*, 2019.
[8] J. Huang, Y. Li, and X. Wang, "Deep gamblers: Learning to abstain with monetary loss," in *ICLR*, 2020.
[9] X. Feng and et al., "Critical examination of selection mechanisms in selective prediction models," *Journal of Machine Learning Research*, 2022.
[10] Łukasz Wołczyk, P. Spurek, T. Trzciński, and J. Tabor, "Adaptive early exits in deep convolutional neural networks," *Journal of Imaging*, 2021.
[11] W. Zhou, C. Xu, T. Ge, J. McAuley, K. Xu, and F. Wei, "Bert loses patience: Fast and robust inference with early exit," in *Advances in Neural Information Processing Systems*, 2020.
[12] X. Liu, T. Sun, J. He, J. Wu, L. Wu, X. Zhang, H. Jiang, Z. Cao, X. Huang, and X. Qiu, "Towards efficient nlp: A standard evaluation and a strong baseline," *arXiv preprint arXiv:2110.07038*, 2021.
[13] M. Wołczyk, B. Wójcik, K. Bałazy, I. T. Podolak, J. Tabor, M. Śmieja, and T. Trzcinski, "Zero time waste: Recycling predictions in early exit neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 2516–2528, 2021.
[14] N. Okati, A. De, and M. Rodriguez, "Differentiable learning under triage," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9140–9151, 2021.
[15] G. Korol, M. G. Jordan, M. B. Rutzig, J. Castrillon, and A. C. S. Beck, "Pruning and early-exit co-optimization for cnn acceleration on fpgas," in *2023 Design, Automation & Test in Europe*. IEEE, 2023, pp. 1–6.