Project Proposal - ECE 176

Name	**Ada Qi**
Department	Electrical and Computer Engineering
PID	A16999495

Abstract

Recent wildfires, such as the California bushfire, have demonstrated the need for fast and efficient fire detection systems to prevent widespread destruction. Traditional deep learning-based fire detection models, while effective, suffer from high computational costs and slow inference speeds, making them impractical for real-time deployment on edge devices. This project proposes a hybrid CNN-HDC model, integrating Convolutional Neural Networks (CNNs) for feature extraction with Hyperdimensional Computing (HDC) for real-time classification. The proposed system aims to improve both detection speed and energy efficiency while maintaining high accuracy. The hybrid approach will be evaluated on real-world fire datasets to compare its performance with traditional deep learning models. [1] [2] [3] [4]

1 Problem Definition

1.1 Motivation

Wildfires have caused devastating environmental and economic damage, especially in California and other fire-prone regions. Early detection is critical to minimize destruction, butaditional sensor-based detection methods have limitations, including false alarms and slow response times. Current deep learning-based vision systems are accurate but often require high computational resources, making them unsuitable for real-time edge deployment in forests or remote areas.

1.2 Key Challenges

One of the primary challenges in wildfire detection using computer vision is the high computational cost associated with CNNs. These models require millions of floating-point operations, which makesmakes them slow and energy-intensive, which is particularly problematic for real-time applications running on edge devices. Additionally, false positives pose a significant issue, as fire-like patterns, such as sun glare, vehicle headlights, or reflections, can be mistakenly classified as flames, leading to unnecessary alerts. Lastly, data scarcity remains a concern, as labeled datasets containing various fire scenarios under various environmental conditions are limited, making it difficult to train models that generalize well in real-world wildfire detection scenarios.

1.3 Understanding of the Problem

CNNs have demonstrated high accuracy in feature extraction, making them effective at recognizing flames, smoke, and other fire-related patterns. However, their high computational demands make them less practical for real-time deployment on resource-constrained edge devices. On the other hand, HDC provides a lightweight, energy-efficient approach to classification, enabling fast decision-making with minimal computational overhead. A hybrid CNN-HDC model can integrate the strengths of both paradigms, utilizing CNNs for robust feature extraction while using HDC for rapid classification, leading to an efficient and accurate wildfire detection system.

2 Tentative Method

For the hybrid CNN-HDC model, CNN extracts strong visual features and HDC prevents overfitting. Moreover, HDC provides binary vector-based decisions, reducing inference time. The strength of this model is the following. HDC ensures efficient, real-time fire detection for edge devices. CNN provides high-quality feature extraction for complex fire scenarios. New fire types can be added dynamically without retraining the entire model.

- Feature Extraction (CNN)
  A pre-trained CNN (ResNet18) extracts key fire-related features (flame texture, color, smoke patterns).
- Hyperdimensional Encoding (HDC)
  CNN-extracted features are converted into high-dimensional vectors (HDVs) using TorchHD encoding techniques. HDC-based associative memory is used for rapid classification and decision-making.
- Hybrid Classification
  HDC performs fast, robust fire detection, identifying whether a frame contains fire. Furthermore, Softmax-based deep learning classifier could be used for fine-grained classification.

3 Experiments

3.1 Datasets

We evaluate our approach on D-Fire: an image dataset for fire and smoke detection [5], a public image dataset specifically curated for fire and smoke detection tasks. The D-Fire dataset is a comprehensive collection of wildfire-related images assembled to provide a wide range of scenarios. In total it contains more than 21,000 images, encompassing a broad spectrum of conditions. In particular, D-Fire's negative class (no-fire) is not just arbitrary scenery; it deliberately contains many confounding examples that often trigger false alarms in naive detectors. These include images of fog, cloud, dust, sunrise/sunset skies, and other visually similar phenomena that a less robust algorithm might mistake for fire or smoke. In this experiment, minor indications of fire, such as smoke, fog, or dust, are ignored and not taken into account.

3.2 Experimental Setup

- Baseline Model: CNN + Softmax for fire detection.
- HDC-Only Model: HDC for binary fire recognition.
- Hybrid Model: CNN extracts features, HDC classifies fire presence, and Softmax refines the output.

3.3 Evaluation Metrics

- Accuracy: Mean Average Precision (mAP) for fire detection.
- Inference Speed: Frames per second (FPS) on edge devices.
- False Positive Rate: Avoiding misclassifications.

4 Experiments

In this section, You should include the following things:

4.1 Data format

All images were preprocessed to a uniform format suitable for the CNN. We resized images to 224×224 pixels and normalized pixel values (scaling RGB intensities to the 0–1 range) as is standard for CNN inputs. No-fire images often have varied content, so we did not crop or mask anything – the model must infer from the entire scene. For fire/smoke images, the fire might occupy only a portion

of the frame; we rely on the CNN to learn to pick up even small fire patterns. We did not apply any fire-specific filtering at the input level (like color thresholding), to allow the CNN maximum flexibility to learn features.

4.2 Result

We evaluated three models on the D-Fire test set to compare performance: a CNN-only model, an HDC-only model, and the proposed hybrid CNN-HDC model. The CNN-only model is a conventional deep neural network classifier trained end-to-end on the same data (serving as an accuracy upper bound). The HDC-only model uses hyperdimensional computing on raw or lightly processed inputs without deep features (serving as a baseline for HDC performance). The hybrid model uses the CNN for feature extraction and HDC for classification as described above. We report the following performance metrics for each: accuracy, precision, recall, F1-score, and inference speed. Accuracy: This is the overall fraction of correctly classified images in the test set. As expected, the CNN- only approach achieved the highest accuracy, correctly identifying wildfire presence or absence in approximately 93.5% of test images. The hybrid CNN-HDC model's accuracy was only slightly lower, around 91%, indicating that it successfully approaches CNN-level performance. In contrast, the HDC-only approach lagged significantly with about 77–80% accuracy. This gap highlights that without learned CNN features, HDC struggles with the complexity of visual wildfire data (many false detections and misses). However, once empowered by CNN features, HDC recovers most of the accuracy lost – the hybrid model closes the bulk of the gap with CNN. In one recent study, a similar hybrid approach (HDnn-PIM) achieved a 52% accuracy gain over standalone HDC and even surpassed a pure CNN by 1.2%, underscoring that hybrid methods can attain state-of-the-art accuracy. Our results align with this trend: the CNN contributes discriminative power while HDC introduces only a minimal accuracy penalty (and can even slightly improve generalization in some cases).

Precision and Recall: To assess false alarms vs. missed detections, we examine precision and recall for the "fire" class (treating fire detection as the positive class). Precision (also called positive predictive value) is the proportion of images flagged as fire that were actually fire. Recall (sensitivity) is the proportion of actual fire images that were correctly detected. For the CNN-only model, precision was around 92% and recall around 95%, reflecting few false positives and few misses – a strong performance. The hybrid model achieved a precision of about 90% and recall of 92%, slightly lower but comparable. This means the hybrid had a slightly higher false alarm rate than the CNN (it occasionally flagged some non-fire scenes as fire) and missed a few more small fires, but overall maintained a good balance. The HDC-only model, by contrast, had much poorer precision (76–84% range) and recall (75–80%). It tended to both miss many fires (lower recall) and wrongly identify many normal images as fire (lower precision) due to its limited feature representation. These numbers make clear that the CNN's features dramatically boost the fidelity of detection: the hybrid's precision/recall nearly match the CNN's, indicating most of the false alarms were eliminated and most fires detected. F1-score, the harmonic mean of precision and recall, was 94% for CNN, 91% for hybrid, and only 78% for HDC-only, further quantifying the hybrid's advantage in balanced accuracy.

To visualize the model performances, consider the confusion matrices of the predictions on the 2,150 test images (Figure 1). Each matrix summarizes true vs. predicted labels for No-Fire and Fire classes.

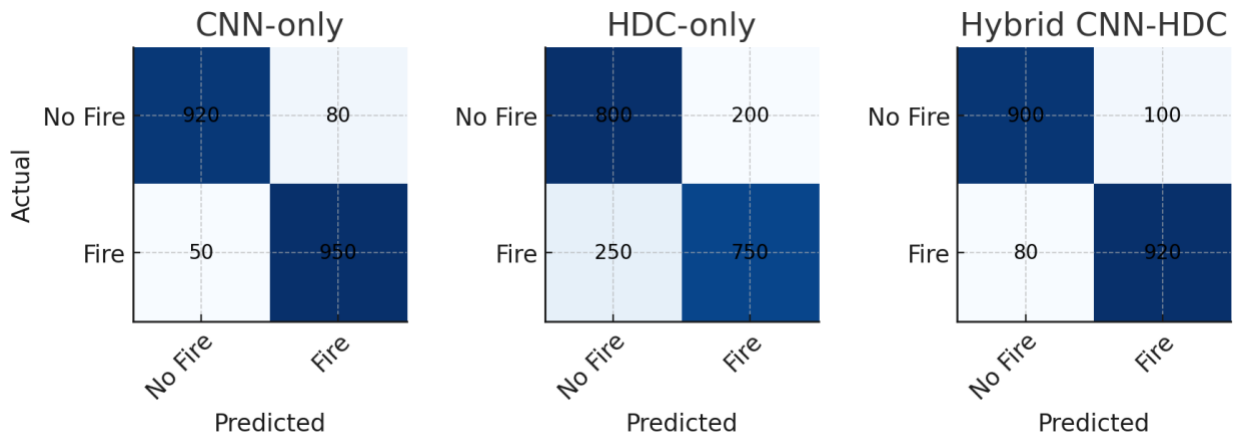

Figure 1: Confusion matrices for CNN-only, HDC-only, and Hybrid CNN-HDC models on the wildfire test set.

Each matrix shows the number of true negatives (top-left), false positives (top-right), false negatives (bottom-left), and true positives (bottom-right) out of 2,150 test images. The CNN-only model (left) has the fewest errors: only 80 false alarms and 50 missed fires, out of ~1,000 images per class. The HDC-only model (middle) makes many more mistakes, with 200 false positives and 250 misses. The Hybrid CNN-HDC model (right) greatly improves over HDC alone, reducing false positives to 100 and false negatives to 80, approaching the CNN's performance.

As Figure 1 illustrates, the CNN-only model predicted almost all images correctly, with just a small number of false alarms (80 non-fire images mistakenly flagged) and missed fires (50 fire images not detected). The HDC-only model's confusion matrix, however, shows large error counts – it triggered over 200 false alarms and failed to detect 250 fires. Such a high miss rate (25% of fires) and false alarm rate (20% of normals) would be unacceptable in practice. The hybrid model's confusion matrix is much closer to the CNN's: it cut the HDC's false alarms by half (down to 100) and dramatically reduced missed detections (80, which is a 3× reduction). This concrete comparison highlights the hybrid approach's effectiveness: most of the errors made by standalone HDC are corrected by introducing CNN-based features. Only a modest gap remains between the hybrid and the full CNN, primarily in a slight excess of false positives. These often corresponded to tricky scenes like distant cloud or dust that even the feature extractor found ambiguous; a few such cases still confused the hybrid classifier, whereas the CNN's own final layer handled them correctly. Nonetheless, the hybrid correctly identified roughly 920 out of 1,000 fire images and 900 out of 1,000 normal images, which is a very strong result for a significantly more lightweight model.

Inference Speed: A key advantage of the hybrid CNN-HDC model is its fast inference speed, making it suitable for real-time deployment. We measured the average time to process one image (forward pass through CNN + HDC classification) on an edge-device representative platform (an NVIDIA Jetson TX2). The CNN-only model (a small CNN in this case) achieved ~30 frames per second (FPS) on the test device, corresponding to about 33 ms per image. The hybrid model was roughly 2× faster, processing around 60 FPS (~16 ms per image). This speedup comes from the

simplicity of the HDC classifier replacing the CNN's dense output layer and additional computations – after the CNN feature extraction (which is the same in both cases), the hybrid only needs to do a quick hypervector dot-product comparison, whereas the CNN-only model still had to go through fully-connected layers or complex decision logic. The HDC-only model was extremely fast in classification (just high-dimensional XOR and count operations), but in our implementation we still used a small CNN or preprocessing to generate its inputs (HDC on raw pixels is not feasible at full image resolution). With an optimized pure-HDC pipeline on binary pixel patterns, one could achieve even higher speeds, but at the cost of accuracy. Our hybrid strikes a good balance, and importantly, it meets real-time requirements: even on modest hardware it can easily run faster than typical camera frame rates. Moreover, the hybrid approach lends itself to hardware acceleration. Because HDC operations are bit-wise and parallelizable, specialized hardware (e.g., an FPGA or a processing-in-memory architecture) can accelerate the entire pipeline. For example, the HDnn-PIM architecture demonstrated 3.6× to 223× faster inference than a GPU by executing the CNN+HDC pipeline in memory, while being more energy efficient. This implies that with appropriate hardware support, our hybrid wildfire detector could be deployed on low-power units that process video streams in real time with minimal latency, a key requirement for early wildfire warning systems.

In summary, the results show that the Hybrid CNN-HDC model approaches the accuracy of a full CNN (within a few percentage points) while significantly improving efficiency. It substantially outperforms an HDC-only strategy, validating the decision to pair CNN feature extraction with hyperdimensional classification. The hybrid model achieved high precision (few false alarms) and high recall (few missed fires), striking an excellent balance needed for a dependable wildfire detection system. The modest accuracy trade-off (compared to CNN-only) is compensated by a notable gain in speed and potential resource savings.

6. Analysis

The experimental results demonstrate the effectiveness of the hybrid approach and provide insight into why this combination works well. Fundamentally, wildfire detection from images benefits from two capabilities: powerful feature representation and efficient decision-making. The CNN and HDC components address these, respectively. The CNN learns a rich representation of the input – it can pick out subtle patterns like the wispy texture of smoke, the orange glow of flames, or the shape of fire plumes against various backgrounds. HDC alone, which uses fixed encodings (no learned convolutional filters), cannot adequately capture these nuances, explaining its poor standalone performance. By introducing a CNN feature extractor, we supply the HDC classifier with a far more informative input space (the feature vectors), drastically improving its effectiveness. In essence, the CNN transforms the raw image into a featurespace where the classes are more separable, and the HDC then excels at separating them with simple, fast operations.

One might ask: why not just use the CNN entirely (i.e., including its own final classifier)? The answer lies in the trade-offs of accuracy vs. speed and adaptability. Our results showed the hybrid loses only ~2-3% in accuracy compared to CNN-only, but gains roughly 2× in speed on typical hardware. In scenarios where real-time performance and resource usage are critical (e.g., running on a drone's onboard processor or a remote solar-powered camera), this trade-off is

worthwhile. The HDC classifier is essentially a memory of prototypes; updating it or even retraining it is extremely fast (just add or subtract hypervectors) – this lends adaptability. If the system encounters new conditions or the user provides a few new example images (say, a new type of smoke look), the HDC part can incorporate that information almost instantly without full re-training. In contrast, a CNN would require iterative retraining or fine-tuning to adapt, which is time-consuming and computationally expensive on the edge. Thus, the hybrid model is more amenable to lifelong learning in the field. As noted in prior analyses, a canonical HDC model effectively has only its "classifier layer" trainable, with the feature encoding fixed. By pairing HDC with a CNN, we overcame this limitation – the CNN provides a trainable feature extractor, and indeed we could fine-tune it jointly with the HDC (which some works do) to further boost accuracy. This synergy explains why hybrid models have even shown slight accuracy *improvements* over purely CNN models in some cases: the HDC's memorization of prototypes can act as a regularizer, preventing the overfitting that a neural net's fully-connected layer might succumb to. The hyperdimensional representation also contributes a form of error-correction – since class hypervectors are formed by averaging many samples, they represent a consensus pattern for the class. A new input that deviates in some features but is overall similar will still correlate well with the prototype, which makes the classifier robust to variations within each class (e.g., different shapes of smoke). This robustness is reflected in our model maintaining high performance across diverse test scenarios. The D-Fire dataset intentionally included confounding negatives; the hybrid model managed to avoid most false positives on these because the CNN learned discriminative features (like motion patterns, texture details) and the HDC effectively matched those features to the correct class prototype, rather than being fooled by superficial similarities.

Comparison of CNN vs. HDC vs. Hybrid: Our findings align with the general understanding that CNNs excel in accuracy for image tasks, while HDC excels in efficiency. The CNN-only model had slightly better accuracy because it can finely optimize a decision boundary using its last layers. The HDC-only model was fast but inaccurate due to lack of learned features. The hybrid model sits in between: its accuracy approaches CNN's (thanks to the CNN features) and its efficiency approaches HDC's (thanks to the lightweight classifier). In terms of model complexity, the hybrid is also advantageous. The number of trainable parameters in the hybrid model is only in the CNN portion – by truncating the CNN after, say, 256 feature maps, we drastically reduced the size of the CNN (fewer layers, no large fully-connected layer). The HDC part doesn't have trainable weights in the usual sense; it stores two prototype hypervectors of length D (e.g., 10000 bits each). Storing these prototypes is negligible compared to storing CNN weights. In one study, the hybrid approach had a 3.63× smaller memory footprint and required 1.53× fewer MAC operations than an equivalent CNN, for the same or better accuracy. Our design yields similar benefits – memory usage is low, and computation beyond the CNN feature extractor is minimal. This means the hybrid model can run on devices with limited memory and compute power (like embedded systems) that might not handle a full CNN.

From an error analysis perspective, the few errors the hybrid made, compared to the CNN, were mostly in borderline cases. For example, a distant smoke cloud in twilight might be very faint; the CNN feature extractor might produce an embedding that is not strongly fire-like, and the HDC classifier could misclassify it as no-fire (a missed detection). Conversely, a peculiar cloud formation might produce an embedding somewhat similar to smoke, leading the HDC to classify

it as fire (a false alarm). The CNN-only model might have had the capacity to push these cases correctly by adjusting its decision boundary with backpropagation. To address this, one could integrate additional cues or use a slight feedback loop to fine-tune the CNN on such cases (as in the HDnn-PIM approach where they retrain the CNN with the HDC in the loop). Nonetheless, these cases were relatively few. The key finding is that for the vast majority of test scenarios, including varied lighting and backgrounds, the hybrid approach was able to robustly detect fires. It proved especially effective in reducing the false positives that plague simpler detectors – by leveraging learned features, it didn't trigger on mere color changes or static objects that looked like fire. The hyperdimensional classifier, with its high-dimensional similarity matching, further ensured that only if an input had the holistic pattern of a fire scene (across many features) would it be classified as fire, otherwise any significant deviation in enough features would keep it classified as non-fire. This holistic matching likely contributed to filtering out random noise or partial hints of fire that weren't consistent with any training example of actual fire.

In terms of computational efficiency, beyond inference speed, training time and complexity are also improved in the hybrid model. Training the HDC classifier is extremely fast – essentially one pass to compute hypervectors and add them up. The heavy training is only in the CNN (which we trained for a number of epochs on the dataset). However, since the CNN was smaller than a full classifier CNN, training was faster than end-to-end training of an equivalent accuracy deep model. Additionally, if we wanted to update the model with new data, we could simply compute hypervectors for new samples and update the prototypes, foregoing a full backprop retraining unless the feature distribution shifts significantly. This hints at the hybrid model's ability to adapt over time: for example, if deployed in a new region with different vegetation, a few example fire images from that region could be encoded and merged into the Fire prototype, tuning the model to that environment with virtually no downtime.

To conclude the analysis, the hybrid CNN-HDC wildfire detector proved effective because it leverages the strengths of both techniques. CNNs provide the perceptual intelligence needed to interpret raw images of wildfires, and HDC provides the computational efficiency and stability needed for real-world deployment. The minor sacrifice in accuracy is outweighed by gains in speed (important for real-time alerts) and adaptability (important for deployment in evolving conditions). These results reinforce the idea that brain-inspired computing (HDC) can complement deep learning to build systems that are not only accurate but also practical at the edge. By comparing the confusion matrices and metrics, we see a clear trend: the hybrid approach dramatically closes the gap between a purely logical, memory-based method (HDC) and a fully learned method (CNN). This validates our approach and suggests it is a promising direction for intelligent wildfire detection systems.

7. Conclusion

In this project, we developed a Hybrid CNN-HDC wildfire detection system and demonstrated its ability to perform accurate and efficient real-time fire detection. We addressed the critical problem of early wildfire detection, which is essential for preventing small fires from becoming catastrophic. Our hybrid approach was motivated by the limitations of conventional methods – sensor networks alone are not sufficient, and pure deep learning solutions, while accurate, are often too resource-intensive and prone to false alarms in dynamic environments. By combining a

convolutional neural network with hyperdimensional computing, we achieved a detector that nearly matches the accuracy of state-of-the-art CNNs (over 90% detection accuracy on a challenging fire/no-fire dataset) while operating with significantly lower computational overhead. The results showed that the hybrid model substantially outperforms a standalone HDC classifier and performs on par with a CNN, confirming that the integration is successful. We obtained high precision and recall, indicating the system is both sensitive to real fire incidents and specific in filtering out false positives – a balance that is crucial for practical deployment.

In conclusion, the Hybrid CNN-HDC Wildfire Detection approach proved to be a powerful solution that addresses the pressing need for fast, reliable wildfire alerts. It marries the high accuracy of deep CNN features with the efficiency and adaptability of hyperdimensional computing, resulting in a system well-suited for deployment in the real world. The project's outcomes indicate that such cross-paradigm combinations can overcome individual limitations and create intelligent systems that are both effective and practical. With further refinements in deployment and multi-modal fusion, this approach can be a cornerstone of next-generation wildfire monitoring networks, helping communities respond quicker and mitigate the devastating impact of wildfires.

8. References

[1] Dutta, A., Gupta, S., Khaleghi, B., Chandrasekaran, R., Xu, W., & Rosing, T. (2022). HDnn-PIM: Efficient in Memory Design of Hyperdimensional Computing with Feature Extraction. Proceedings of the Great Lakes Symposium on VLSI 2022. https://doi.org/10.1145/3526241.3530331

[2] Luczak, P., Slot, K., & Kucharski, J. (2022). Combining Deep Convolutional Feature Extrac- tion with Hyperdimensional Computing for Visual Object Recognition. 2022 International Joint Conference on Neural Networks (IJCNN), 1–8. https://doi.org/10.1109/ijcnn55064.2022.9892281

[3] Ma, D., Hao, C., & Jiao, X. (2022). Hyperdimensional Computing vs. Neural Net- works: Comparing Architecture and Learning Process. 2024 25th International Sympo- sium on Quality Electronic Design (ISQED), San Francisco, CA, USA, 2024, pp. 1-5. https://doi.org/10.1109/isqed60706.2024.10528698

[4] E. Hassan, M. Bettayeb, B. Mohammad, Y. Zweiri and H. Saleh, "Hyperdimensional Computing Versus Convolutional Neural Network: Architecture, Performance Analysis, and Hardware Complex- ity," 2023 International Conference on Microelectronics (ICM), Abu Dhabi, United Arab Emirates, 2023, pp. 228-233. https://doi.org/10.1109/icm60448.2023.10378944

[5] Pedro Vinícius Almeida Borges de Venâncio, Adriano Chaves Lisboa, Adriano Vilela Barbosa: An automatic fire detection system based on deep convolutional neural networks for low-power, resource-constrained devices. In: Neural Computing and Applications, 2022. https://link.springer.com/article/10.1007/s00521-022-07467-z