

# Comparative Study of Deep and Hybrid Models for Alzheimer’s Disease Stage Classification Using MRI

Piyush Mehta

*Department of Computer Engineering  
Sardar Patel Institute of Technology  
Mumbai, Maharashtra, India  
piyush.mehta22@spit.ac.in*

Shardul Mahindrakar

*Department of Computer Engineering  
Sardar Patel Institute of Technology  
Mumbai, Maharashtra, India  
shardul.mahindrakar22@spit.ac.in*

Krrish Nichanii

*Department of Computer Engineering  
Sardar Patel Institute of Technology  
Mumbai, Maharashtra, India  
krrish.nichanii22@spit.ac.in*

Dr. Sudhir Dhage

*Department of Computer Engineering  
Sardar Patel Institute of Technology  
Mumbai, Maharashtra, India  
sudhir\_dhage@spit.ac.in*

**Abstract**—Alzheimer’s Disease (AD) poses an escalating global health burden, and timely, stage-accurate diagnosis is essential for effective clinical management. Because the structural brain changes that mark disease progression are gradual and visually subtle, manual interpretation of Magnetic Resonance Imaging (MRI) is both time-intensive and prone to inter-rater variability. This paper addresses that challenge by conducting a systematic, head-to-head evaluation of multiple computational paradigms — custom deep learning, transfer learning, classical machine learning, and hybrid combinations thereof — on a four-class AD staging task covering Non-Demented, Very Mild Demented, Mild Demented, and Moderate Demented conditions. Working with 6,400 preprocessed axial MRI scans, we benchmarked ten distinct models spanning a custom Convolutional Neural Network (CNN), seven pre-trained transfer learning architectures (VGG16, VGG19, ResNet50, InceptionV3, EfficientNetB0, DenseNet169, MobileNetV2), a PCA-SVM pipeline, and deep-feature hybrid classifiers. Our purpose-built CNN topped all competitors, recording a test accuracy of 99.1% alongside an AUC of 0.999 — results that demonstrate the advantage of designing an architecture around the specific characteristics of the target imaging modality rather than relying solely on generic pre-trained weights.

**Index Terms**—Alzheimer’s Disease, MRI, Deep Learning, Convolutional Neural Network (CNN), Transfer Learning, Machine Learning, Image Classification

## I. INTRODUCTION

Alzheimer’s Disease stands among the most disabling of neurodegenerative conditions worldwide, progressively eroding memory, cognition, and functional independence across millions of patients. Its hallmark pathology — diffuse cortical atrophy and hippocampal volume loss — manifests over years before overt clinical symptoms emerge, making the window for intervention narrow once a diagnosis is finally confirmed. Brain MRI has become a cornerstone imaging tool for tracking these structural changes, yet the degree to which atrophy can be reliably quantified depends heavily on the expertise of the interpreting clinician and the consistency of the evaluation protocol. Automating this interpretation using modern machine

learning offers a route toward faster, more reproducible staging.

Within the machine learning landscape, Convolutional Neural Networks occupy a privileged position for image-based tasks because their architectural design — stacked spatial filters followed by pooling — mirrors the hierarchical way visual information is organized. This property allows CNNs to extract discriminative patterns directly from raw pixel intensities without requiring hand-engineered feature descriptors. Transfer learning extends this advantage further: networks pre-trained on millions of natural images can be repurposed for medical imaging applications, contributing powerful low-level feature detectors even when labeled medical data are scarce.

This study undertakes a broad comparative investigation spanning four methodological families. Rather than focusing on a single architecture, we systematically trained and evaluated a custom CNN, seven well-established transfer learning backbones, a dimensionality-reduction pipeline based on PCA combined with traditional classifiers, and hybrid systems that pair deep feature extraction with standalone machine learning classifiers. The objective is to identify which approach best captures the subtle inter-stage differences in MRI-based AD classification and to offer the research community a clear, reproducible benchmark across paradigms. In doing so, we hope to inform both clinical tool development and the selection of architectural strategies in future AD imaging studies.

## II. LITERATURE REVIEW

The conceptual foundation underlying computational approaches to AD diagnosis shifted considerably following the publication of the NIA-AA Research Framework by Jack Jr. *et al.* [1]. That framework redefined disease presence in terms of measurable biological markers — including MRI-derived indicators of brain atrophy — rather than purely clinical observation. From a computational perspective, this re-framing was consequential: it provided an objective, image-grounded target against which algorithmic methods could be trained and

validated, paving the way for quantitative, data-driven staging systems.

Early deep learning contributions to neuroimaging came largely from tasks adjacent to AD classification. Kamnitsas *et al.* [12] demonstrated that a multi-scale 3D CNN operating at two spatial resolutions could segment brain lesions from volumetric MRI with high accuracy, establishing that deep architectures can extract spatially coherent, pathology-relevant features from complex neuroimaging inputs. Although their task differed from ours, the insight — that end-to-end feature learning from MRI avoids the fragility of manually constructed descriptors — directly motivates the CNN-centric approach we adopt. This sentiment was empirically reinforced by Nanni *et al.* [9], whose comparative experiments showed that features learned entirely by a network without human intervention frequently outperform carefully crafted hand-designed alternatives on vision benchmarks.

When labeled data are limited, transfer learning becomes the dominant strategy for deploying deep networks in medical settings. Shin *et al.* [4] provided an early, thorough treatment of this technique in the context of computer-aided detection, examining how fine-tuning ImageNet-pretrained CNNs on radiological images generalizes across acquisition conditions and anatomical regions. Why this generalization works was examined theoretically by Yosinski *et al.* [10], who showed through layer-by-layer experiments that shallow network layers converge to general-purpose edge and texture detectors regardless of domain, while deeper layers become progressively task-specific. This layered generality is precisely what makes pre-trained backbones useful as starting points for specialized medical classifiers.

The backbone architectures evaluated in our study span roughly a decade of innovation. VGGNet [5] established that stacking many small ( $3\times 3$ ) filters to considerable depth reliably improves discriminative power, though its parameter count makes training computationally intensive. He *et al.* [6] addressed the optimization difficulty that accompanies great depth by introducing residual connections, which create direct gradient paths between distant layers and allow networks exceeding a hundred layers to converge reliably. Szegedy *et al.* [7] took a different route by widening each layer through parallel filter banks of multiple sizes, enabling the network to simultaneously attend to features at coarse and fine scales without a proportional increase in parameters. Tan and Le [8] later proposed a principled scaling law — compound scaling — that jointly and proportionally increases a baseline network’s depth, width, and input resolution, yielding the EfficientNet family whose accuracy-to-compute ratio set a new standard.

Several practical challenges arise when deploying these powerful architectures on medical data. Class imbalance — an intrinsic property of most disease datasets, where healthy or early-stage cases far outnumber advanced cases — can mislead a network into prioritizing majority-class accuracy at the expense of minority-class sensitivity. Johnson and Khoshgoftaar [2] catalogued the range of mitigation strategies,

from oversampling minority classes to modifying the loss function, providing practitioners with a structured overview of available remedies. On the optimization side, the Adam algorithm [3] has emerged as the default choice for training deep classification networks, owing to its ability to maintain per-parameter adaptive learning rates that cope gracefully with sparse or noisy gradient signals. Finally, clinical adoption of any algorithmic decision-aid requires that clinicians understand why a model reaches its conclusions. The Grad-CAM visualization technique developed by Selvaraju *et al.* [11] addresses this by backpropagating gradient information to the final convolutional layer and producing a spatial heatmap that highlights the image regions most influential to the prediction, giving radiologists a direct means of verifying model behavior against domain knowledge.

Our work builds on this body of prior art by constructing a unified experimental framework in which all four paradigms — custom deep learning, transfer learning, classical machine learning, and hybrid models — are evaluated under identical data conditions, enabling fair, direct comparison.

### III. METHODOLOGY

The experimental pipeline, depicted in Fig. 1, begins with dataset preparation and proceeds through four independent modeling tracks before converging on a common evaluation stage. Each track is described in full below.

#### A. Dataset and Preprocessing Pipeline

All experiments use the publicly available Alzheimer’s MRI Dataset, comprising 6,400 two-dimensional axial brain scans captured in grayscale and made available in pre-processed form.

1) *Dataset Characteristics:* Each scan was resized to  $128\times 128$  pixels prior to release. The dataset partitions into four disease stages illustrated in Fig. 2:

- **Non-Demented (ND):** 3,200 images
- **Very Mild Demented (VMD):** 2,240 images
- **Mild Demented (MD):** 896 images
- **Moderate Demented (MOD):** 64 images

This distribution is highly skewed: the Non-Demented class accounts for half the dataset, while the Moderate Demented class contributes fewer than one percent of samples. Managing this imbalance during training was therefore an important design consideration.

2) *Data Preparation Pipeline:* Three steps constitute the preparation pipeline:

- 1) **Stratified splitting:** The full set was divided into training (80%), validation (10%), and test (10%) subsets. Stratification was enforced so that each subset preserves the overall class proportions, and a fixed random seed was used throughout to ensure reproducibility.
- 2) **Online augmentation (training set only):** To reduce overfitting without artificially inflating the dataset size through offline duplication, stochastic augmentation was applied at load time exclusively to training batches. Three transformations were employed:

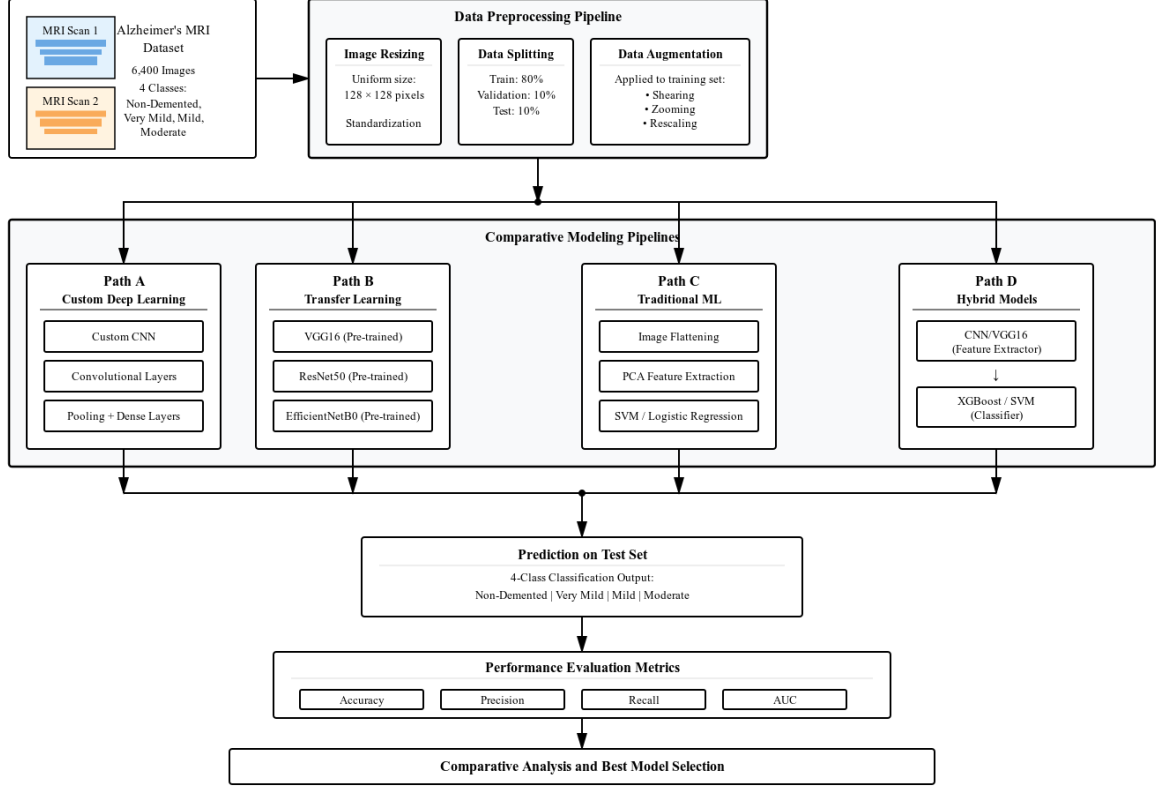


Fig. 1: End-to-end workflow of the comparative study: MRI images enter a preprocessing pipeline before being routed through four parallel modeling tracks whose outputs are consolidated for final performance evaluation.

- **Intensity normalization:** Raw pixel values spanning  $[0, 255]$  were mapped to the unit interval:

$$I_{\text{norm}} = \frac{I_{\text{orig}}}{255.0} \quad (1)$$

- **Random shear:** A shear factor sampled uniformly from  $[-0.2, 0.2]$  was applied along both axes, slightly distorting the image geometry to discourage reliance on rigid spatial cues:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & sh_x \\ sh_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2)$$

- **Random zoom:** Scale factors in the range  $[0.8, 1.2]$  were drawn randomly, resizing images up or down before re-cropping to  $128 \times 128$ .

- 3) **Evaluation normalization:** Validation and test images were subjected only to intensity normalization (Eq. 1); no geometric perturbations were applied, ensuring that reported metrics reflect performance on unmodified data.

### B. Custom Convolutional Neural Network Architecture

Rather than adapting an off-the-shelf backbone, we constructed a lightweight CNN from scratch whose receptive field and parameter budget were calibrated to the  $128 \times 128$  input resolution and the relatively small dataset size. The network

is organized into a convolutional feature extractor followed by a fully connected classification head.

1) *Layer Specification:* Table I enumerates every layer together with its parameters and output tensor shape.

TABLE I: Layer-by-Layer Specification of the Custom CNN Model

| Layer Type                     | Parameters                     | Output Shape         |
|--------------------------------|--------------------------------|----------------------|
| InputLayer                     | Input Shape: (128, 128, 3)     | (None, 128, 128, 3)  |
| <b>Feature Extraction Base</b> |                                |                      |
| Conv2D                         | 16 filters, kernel (3,3), ReLU | (None, 126, 126, 16) |
| MaxPooling2D                   | Pool size (2,2)                | (None, 63, 63, 16)   |
| Conv2D                         | 32 filters, kernel (3,3), ReLU | (None, 61, 61, 32)   |
| MaxPooling2D                   | Pool size (2,2)                | (None, 30, 30, 32)   |
| Conv2D                         | 64 filters, kernel (3,3), ReLU | (None, 28, 28, 64)   |
| MaxPooling2D                   | Pool size (2,2)                | (None, 14, 14, 64)   |
| <b>Classification Head</b>     |                                |                      |
| Flatten                        | —                              | (None, 12544)        |
| Dense                          | 128 units, ReLU                | (None, 128)          |
| Dropout                        | Rate = 0.5                     | (None, 128)          |
| Dense (Output)                 | 4 units, Softmax               | (None, 4)            |

2) *Core Mathematical Operations:* Four operations underpin the CNN's learning process:

- 1) **Discrete convolution:** Each convolutional layer computes a feature map  $F$  by sliding a learned kernel  $K$  across the input  $I$ . The value at spatial position  $(i, j)$

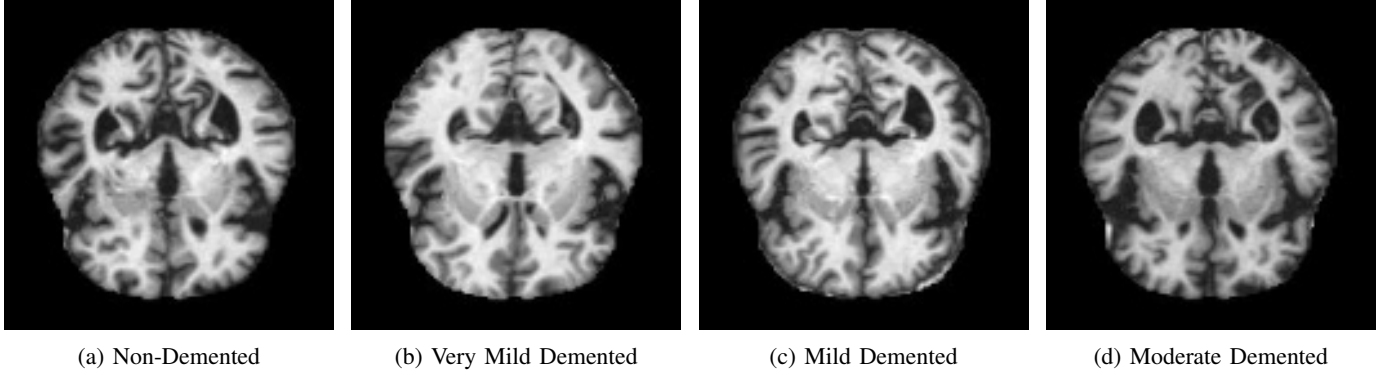


Fig. 2: Representative axial MRI slices from each class, illustrating the progressive cortical thinning and ventricular enlargement that characterize advancing AD.

accumulates a weighted sum of the overlapping input patch:

$$F(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i-m, j-n) K(m, n) \quad (3)$$

Repeating this across multiple kernels allows the layer to detect a variety of local structures simultaneously.

- 2) **Rectified Linear Unit (ReLU)**: After each convolution, the activation

$$\text{ReLU}(x) = \max(0, x) \quad (4)$$

discards negative activations, introducing the non-linearity required for the network to approximate complex, non-linear decision boundaries.

- 3) **Spatial max-pooling**: A  $2 \times 2$  pooling window scans the activated feature map and retains only the largest value in each region:

$$y = \max_{i,j \in R} x_{i,j} \quad (5)$$

This halves the spatial resolution at each stage, compressing representation size while preserving the most salient activations and conferring limited translation invariance.

- 4) **Softmax output**: The final dense layer maps an unconstrained score vector  $z \in \mathbb{R}^C$  (with  $C=4$ ) to a valid probability simplex. The predicted probability assigned to class  $i$  is:

$$\text{Softmax}(z)_i = \frac{e^{z_i}}{\sum_{j=1}^C e^{z_j}} \quad (6)$$

The class index carrying the maximum probability constitutes the model's prediction.

Network weights were optimized end-to-end using the Adam optimizer [3] with Categorical Cross-Entropy as the loss criterion, which is the standard choice for mutually exclusive multi-class problems. A dropout layer with retention probability 0.5 was inserted between the two dense layers to regularize the classification head during training.



Fig. 3: Schematic of the custom CNN: three convolutional blocks progressively double filter depth while halving spatial dimensions, feeding into a regularized two-layer classification head.

### C. Transfer Learning Models

Transfer learning was evaluated as a complementary strategy to custom design. In this paradigm a backbone network — pretrained on a large natural-image corpus — is repurposed as a fixed feature extractor by attaching a new, trainable classification head while keeping the original weights frozen.

1) **Adaptation Protocol**: Each pretrained backbone was adapted through two consistent steps:

- 1) **Loading the convolutional base**: All backbone weights were initialized from ImageNet pretraining. The top classification layers supplied by each model were removed (`include_top=False`), retaining only the convolutional stages that produce high-dimensional spatial feature maps.
- 2) **Attaching a task-specific head**: All backbone layers were frozen to preserve the pretrained representations. A new, trainable classification head — comprising a Flatten operation, a Dense layer with ReLU activation, a Dropout layer, and a four-way Softmax output layer — was stacked on top. Only this head was updated during training on the AD dataset.

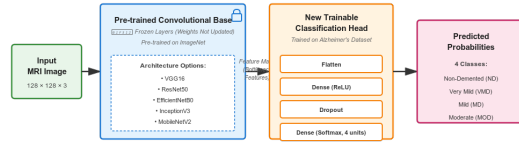


Fig. 4: Transfer learning strategy: the frozen pretrained base acts as a static feature extractor; only the newly attached classification head is trained on the target dataset.

2) *Evaluated Architectures*: Seven backbones spanning different design philosophies were selected:

- **VGG16 and VGG19** [5]: Depth-first architectures from Oxford's Visual Geometry Group that achieve representational richness by stacking sequences of small-kernel ( $3 \times 3$ ) convolutions, trading computational cost for simplicity.
- **ResNet50** [6]: Introduces identity shortcut connections that route information directly across one or more layers. These connections stabilize gradient flow in very deep networks and were the key innovation that made training networks of 50–150 layers tractable.
- **InceptionV3** [7]: Employs parallel branches of differently sized filters within each module, allowing the network to aggregate spatial context at multiple granularities in a single forward pass.
- **EfficientNetB0** [8]: Derived from a compound scaling procedure that simultaneously increases network depth, width, and input resolution according to a fixed ratio, balancing accuracy against parameter count more efficiently than manual scaling.
- **DenseNet169**: Connects every layer to all subsequent layers, maximizing gradient accessibility throughout the network and promoting aggressive feature reuse that reduces redundancy.
- **MobileNetV2**: Replaces standard convolutions with depth-wise separable operations, drastically reducing parameter count; designed for deployment on constrained hardware.

#### D. Machine Learning and Hybrid Models

Two additional paradigms were benchmarked to assess whether deep architectures are strictly necessary for this task.

1) *PCA with Traditional Classifiers*: This pipeline reduces each image to a compact feature vector via Principal Component Analysis before applying conventional classifiers. Working directly from raw pixels would be impractical because a  $128 \times 128 \times 3$  image expands to a 49,152-dimensional vector, making distance-based methods ill-conditioned. The pipeline proceeds as follows:

- 1) **Flattening**: Each image is unrolled into a one-dimensional vector of length 49,152.
- 2) **PCA compression**: The covariance structure of the training vectors is decomposed to identify the orthogonal

directions of maximum variance (principal components). Projecting onto a leading subset of these components discards low-variance, noisy dimensions while retaining the most discriminative signal.

3) **Classification**: Three classifiers were trained on the PCA-projected features:

- **Support Vector Machine (SVM)**: Seeks the maximum-margin hyperplane separating classes in the projected space.
- **Logistic Regression**: A linear probabilistic classifier providing a lower-complexity baseline.
- **Random Forest**: An ensemble of decision trees whose predictions are aggregated by majority vote.

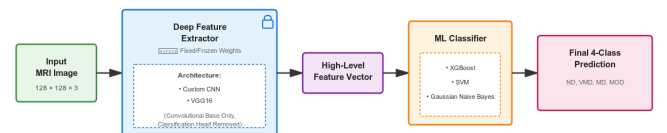
2) *Hybrid Deep-Feature Classifiers*: The hybrid approach decouples representation learning from classification. A pre-trained network (either the custom CNN or VGG16) with its softmax head removed is used solely to map each input image to a dense feature vector extracted from its penultimate layer. These vectors encode high-level semantic content derived from the network's learned filters and are then used as input to standalone classifiers:

- 1) **Feature extraction**: Images are passed through the truncated network; the output tensor from the last pooling or flatten layer serves as the image's feature descriptor.
- 2) **Classifier training**: Three classifiers were evaluated on the extracted features:

- **XGBoost**: A gradient-boosted ensemble that iteratively fits new trees to the residual errors of the current ensemble, offering strong performance on tabular and feature-vector inputs.
- **SVM**: Operates on the deep features rather than PCA-reduced pixels, testing whether richer representations improve SVM performance.
- **Gaussian Naive Bayes (GNB)**: A probabilistic baseline that models each feature dimension independently under a Gaussian assumption.



(a) PCA-based machine learning pipeline.



(b) Hybrid model using deep feature extraction.

Fig. 5: Processing pipelines for the traditional and hybrid modeling tracks.

#### IV. RESULTS AND ANALYSIS

All models were assessed on the held-out test set using five metrics: Test Loss, Accuracy, AUC, Precision, and Recall. This multi-metric view is important because accuracy alone can be misleading under class imbalance; AUC and recall provide complementary information about a model’s discrimination and sensitivity across all four stages.

##### A. Deep Learning and Transfer Learning Performance

Table II consolidates results for the custom CNN and all seven transfer learning backbones. The majority of deep models surpassed 85% accuracy, confirming that learned spatial representations are well-suited to this classification problem.

The custom CNN achieved the strongest results on every metric: 99.1% accuracy, an AUC of 0.999, and a test loss of just 0.035. Its margin over the next-best deep model — EfficientNetB0 at 96.2% — suggests that a purpose-designed architecture with comparatively few parameters can outperform larger, more general backbones when matched to the target domain. MobileNetV2 underperformed sharply, reaching only 58.1% accuracy; its compact design appears insufficient to capture the fine-grained atrophy patterns that differentiate AD stages in this dataset.

##### B. Machine Learning Model Performance

Table III reports results for the PCA-based classifiers.

The SVM trained on PCA features attained 99.0% accuracy — a result that rivals the custom CNN and substantially exceeds all transfer learning models. This outcome is notable because it demonstrates that a classical classifier can match state-of-the-art deep learning when paired with an effective dimensionality reduction step. PCA appears to isolate a compact, linearly separable representation of the inter-class variance that SVM exploits with high efficiency. Random Forest, by contrast, achieved only 71%, indicating that tree-based ensembles are less suited to the high-dimensional PCA feature space than margin-based methods.

##### C. Comparative Analysis

Fig. 6 plots accuracy across all tested models. Two approaches separate clearly from the rest: the custom CNN and the PCA-SVM pipeline, both converging near 99%.

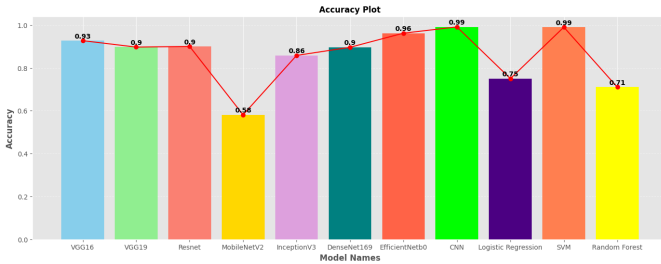


Fig. 6: Test accuracy of all implemented models. The custom CNN and PCA-SVM pipeline form a distinct top tier, outperforming all transfer learning backbones.

This convergence of two methodologically distinct paradigms at the same accuracy level is the central finding of our study. It implies that high performance on this dataset does not require the representational depth of a large pretrained model — provided that the chosen method is well-matched to the data. For the custom CNN, that match comes from an architecture scaled to  $128 \times 128$  inputs without the excess capacity that makes fine-tuning large backbones unstable on small datasets. For PCA-SVM, the match comes from PCA’s ability to compress pixel space into directions aligned with the dominant visual variance between AD stages, giving SVM a well-conditioned optimization landscape.

The transfer learning models occupy a middle tier between approximately 85% and 96%. Their pretraining on natural photographic images likely introduces a representation bias — filters tuned to color, texture, and object edges — that does not fully align with the grayscale atrophy gradients relevant to AD staging. Fine-tuning only the classification head, while computationally convenient, may be insufficient to overcome this domain gap.

#### V. CONCLUSION

This work presented a structured comparative study spanning ten distinct models across four computational paradigms for four-class Alzheimer’s Disease stage classification from brain MRI. Our purpose-built CNN, designed around the input resolution and class structure of the target dataset, achieved 99.1% test accuracy and an AUC of 0.999 — outperforming every evaluated transfer learning backbone. Equally noteworthy was the PCA-SVM pipeline’s 99.0% accuracy, demonstrating that classical methods retain competitive power when an appropriate dimensionality reduction precedes classification. Together, these findings challenge the assumption that deep pretrained backbones are universally superior in medical imaging tasks and highlight the value of matching model capacity to dataset scale and domain.

Future directions for this research include expanding to three-dimensional MRI volumes with 3D-CNN architectures, incorporating longitudinal data to track disease progression over time, and integrating complementary clinical biomarkers to build multimodal diagnostic systems. Crucially, any model intended for clinical deployment should be subjected to rigorous interpretability analysis — for instance through gradient-based attribution maps such as Grad-CAM [11] — to verify that predictions are grounded in radiologically meaningful image features rather than spurious correlations.

#### REFERENCES

- [1] C. R. Jack Jr. et al., “NIA-AA Research Framework: Toward a biological definition of Alzheimer’s disease,” *Alzheimer’s & Dementia*, vol. 14, no. 4, pp. 535–562, 2018, doi: 10.1016/j.jalz.2018.02.018.
- [2] J. M. Johnson and T. M. Khoshgoftaar, “Survey on deep learning with class imbalance,” *Journal of Big Data*, vol. 6, no. 27, 2019, doi: 10.1186/s40537-019-0192-5.
- [3] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

TABLE II: Performance Metrics of Deep Learning Models on the Test Set

| Model          | Test Loss    | Test Accuracy | Test AUC     | Test Precision | Test Recall  |
|----------------|--------------|---------------|--------------|----------------|--------------|
| VGG16          | 0.193        | 0.927         | 0.993        | 0.928          | 0.924        |
| VGG19          | 0.279        | 0.897         | 0.986        | 0.898          | 0.891        |
| ResNet         | 0.324        | 0.900         | 0.982        | 0.903          | 0.897        |
| MobileNetV2    | 0.941        | 0.581         | 0.842        | 0.620          | 0.483        |
| InceptionV3    | 0.426        | 0.858         | 0.974        | 0.867          | 0.852        |
| DenseNet169    | 0.304        | 0.896         | 0.981        | 0.899          | 0.889        |
| EfficientNetB0 | 0.110        | 0.962         | 0.997        | 0.964          | 0.962        |
| <b>CNN</b>     | <b>0.035</b> | <b>0.991</b>  | <b>0.999</b> | <b>0.991</b>   | <b>0.991</b> |

TABLE III: Performance Metrics of Machine Learning Models on the Test Set

| Model               | Accuracy    | Precision   | Recall      |
|---------------------|-------------|-------------|-------------|
| Logistic Regression | 0.75        | 0.81        | 0.73        |
| <b>SVM</b>          | <b>0.99</b> | <b>0.99</b> | <b>0.99</b> |
| Random Forest       | 0.71        | 0.64        | 0.42        |

- [4] H. C. Shin et al., "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1285–1298, May 2016, doi: 10.1109/TMI.2016.2528162.
- [5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [7] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE CVPR*, 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.
- [8] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th ICML*, 2019, vol. 97, pp. 6105–6114.
- [9] L. Nanni, S. Ghidoni, and S. Brahnam, "Handcrafted vs. non-handcrafted features for computer vision," *Pattern Recognition*, vol. 71, pp. 158–178, 2017, doi: 10.1016/j.patcog.2017.05.025.
- [10] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Adv. Neural Inf. Process. Syst. 27 (NIPS)*, 2014, pp. 3320–3328.
- [11] R. R. Selvaraju et al., "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE ICCV*, 2017, pp. 618–626, doi: 10.1109/ICCV.2017.74.
- [12] K. Kamnitsas et al., "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation," *Med. Image Anal.*, vol. 36, pp. 61–78, 2017, doi: 10.1016/j.media.2016.10.004.