Parameter–Efficient MobileViT: A Modular Exploration of Lightweight Convolution, Linear Attention, and Fusion Strategies

A Thesis Submitted
In Partial Fulfillment of the Requirements for the Degree of

Bachelors of Technology
in
Information Technology

Samyak Jain	**IIT2021104**
Atharva Gadekar	**IIT2021049**
Anshul Bhardwaj	**IIT2021057**

Under the Supervision of
Prof. Pritish Kumar Varadwaj

to the

DEPARTMENT OF INFORMATION TECHNOLOGY

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY ALLAHABAD, INDIA

May 2025

CANDIDATE DECLARATION

We hereby declare that the work presented in this report entitled **"Parameter–Efficient MobileViT: A Modular Exploration of Lightweight Convolution, Linear Attention, and Fusion Strategies"**, submitted towards fulfillment of BACHELOR'S THESIS report in Department of Information Technology at Indian Institute of Information Technology Allahabad, is an authenticated record of our original work carried out under the guidance of **Prof. Pritish Kumar Varadwaj**. Due acknowledgements have been made in the text to all other material used. The project was done in full compliance with the requirements and constraints of the prescribed curriculum.

Anirudh Arora
IIT2019003
Information Technology

Pratyush Pareek
IIT2019184
Information Technology

Muskan Deep Kaur Maini
IEC2019088
Electronics & Communication Engineering

CERTIFICATE FROM SUPERVISOR

This is to certify that the statement made by the candidate is correct to the best of my knowledge and belief. The project titled **Traceback of Data Poisoning Attacks on Federated Learning Systems** is a record of candidates' work carried out by them under my guidance and supervision. I do hereby recommend that it should be accepted in the fulfillment of the requirements of the Bachelor's thesis at IIIT Allahabad.

Prof. O.P. Vyas

CERTIFICATE OF APPROVAL

The forgoing thesis is hereby approved as a creditable study carried out in the area of Information Technology and presented in a manner satisfactory to warrant its acceptance as a pre-requisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the thesis only for the purpose for which it is submitted.

Committee on final examination for the evaluation of thesis:

1. Supervisor Name _____ _____
2. Board Member 1 _____ _____
3. Board Member 2 _____ _____

_____
Dean(A & R)

Acknowledgement

We take this opportunity to express my profound gratitude and deep regards to my guide **Prof. Pritish Kumar Varadwaj, Department (IT), IIIT-Allahabad**, for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him from time to time shall carry us a long way in the journey of life on which we are about to embark.

Contents

List of Figures

List of Tables

Abstract

Deploying vision transformers on mobile and embedded devices demands a careful balance between model expressiveness and computational efficiency. This thesis presents a systematic ablation study of the *MobileViT* block, replacing each of its three constituent modules—convolution, self-attention, and feature fusion—with lightweight alternatives culled from the literature. A full factorial grid comprising four convolutional operators (standard, depthwise, grouped, pointwise), three attention mechanisms (multi-head, separable, low-rank), and three fusion strategies (concat+3×3, concat+1×1, residual add) yields thirty-six candidate architectures, fourteen of which are exhaustively trained on CIFAR-10. Two variants emerge as Pareto-optimal: (i) **standard convolution + separable attention + residual fusion**, achieving 80.32 % accuracy with 45 % fewer parameters and 25 % faster training time than the baseline; and (ii) **grouped convolution + low-rank attention + residual fusion**, attaining 79.29 % accuracy at a 36 % parameter reduction. These findings demonstrate that quadratic self-attention and heavy fusion convolutions are not prerequisites for competitive accuracy in lightweight transformer–CNN hybrids. The modular code-base released with this work enables rapid prototyping of further variants and lays the groundwork for automated architecture search under strict mobile constraints.

Chapter 1

Introduction

1.1 Introduction

Vision transformers have recently demonstrated remarkable performance on a variety of image recognition tasks, challenging the long-standing dominance of convolutional neural networks.

However, pure transformer architectures tend to be large and computationally expensive, making them ill-suited for deployment on resource-constrained devices such as mobile phones and embedded systems.

To bridge this gap, hybrid architectures that integrate convolutional inductive biases with self-attention mechanisms have been proposed, achieving a balance between accuracy and efficiency. Among these, a lightweight transformer block that interleaves local convolution and global self-attention—hereafter referred to as the *MobileViT block*—offers a promising direction for mobile vision.

Despite its merits, the baseline MobileViT block still incurs a non-negligible parameter and computational overhead when scaled to real-world tasks. Furthermore, each of its three core components—(1) the convolutional stem, (2)

the multi-headed self-attention module, and (3) the fusion layer that merges local and global features—represents an opportunity for optimization. By systematically exploring alternative designs for each component, it may be possible to significantly reduce the overall parameter count and inference latency while preserving, or even improving, recognition accuracy.

In this work, we conduct a comprehensive ablation study over the *design space* of the MobileViT block. Specifically, we evaluate:

- **Convolutional variants**, including depthwise separable convolutions, grouped convolutions, and pure pointwise (1×1) convolutions;
- **Attention mechanisms**, covering standard multi-head self-attention, linear-complexity separable attention, and low-rank attention approximations;
- **Fusion strategies**, from the original concatenation-and-3×3-convolution to lighter alternatives such as concatenation-and-1×1-convolution or residual add-only fusion.

Our experiments span a full factorial grid of these variants, resulting in thirty-six model configurations. We benchmark each on CIFAR-10—measuring top-1 accuracy, parameter count, and average epoch runtime—to identify Pareto-optimal trade-offs. The results yield clear guidelines for selecting component variants that minimize model size and latency with minimal impact on accuracy.

1.2 Objectives

The primary objective of this thesis is to design and evaluate a parameter-efficient variant of the MobileViT block that delivers near-baseline accuracy

with substantially reduced model size and training/inference time. Specifically, we aim to:

1. **Develop a modular implementation** of the MobileViT block that allows seamless substitution of its three core components—convolution, attention, and fusion—with alternative, lightweight techniques.
2. **Systematically explore** the design space by instantiating all combinations of four convolution variants (standard, depthwise, grouped, pointwise), three attention mechanisms (multi-head, separable, low-rank), and three fusion strategies.
3. **Benchmark each configuration** on the CIFAR-10 dataset, recording top-1 accuracy, parameter count, and average epoch runtime, in order to identify Pareto-optimal trade-offs between model compactness and predictive performance.
4. **Select the best performing configuration** under a strict parameter budget (around 1 million parameters) and verify that its accuracy drop relative to the original MobileViT block does not exceed 1 percentage point.
5. **Compare the optimized model** directly against the baseline MobileViT implementation, highlighting improvements in parameter efficiency and runtime while quantifying any shifts in accuracy.

Through these objectives, we aim not only to produce a more efficient MobileViT variant suitable for deployment on resource-constrained devices but also to derive general design guidelines for lightweight transformer–CNN hybrids in mobile vision.

Chapter 2

Literature review

2.1 Howard *et al.*: MobileNets[1]

This paper introduces *depthwise–separable convolutions*, splitting a standard 3×3 convolution into depthwise and pointwise stages to achieve a 9× reduction in parameters and computation compared with dense kernels.

They demonstrate that, with proper width and resolution multipliers, the resulting network family attains competitive ImageNet accuracy while retaining real-time performance on mobile CPUs.

Insight adopted. The depthwise variant of our convolution block is implemented exactly in this spirit, providing a low-cost spatial operator that serves as the baseline for all subsequent MobileViT variants in our study.

2.2 Sandler *et al.*: MobileNetV2[2]

This paper proposes the *inverted residual bottleneck* architecture, where a narrow input tensor is first expanded by a 1×1 convolution, processed by a depth-

wise spatial filter, and finally projected back to a low-dimension representation.

They argue that performing expensive spatial operations in a high-dimensional manifold while shortcutting the narrow tensor drastically decreases parameters yet preserves representational power.

Insight adopted. Our backbone retains this bottleneck structure outside the MobileViT blocks; moreover, the pure `pointwise` convolution option in the ablation grid reflects the MobileNetV2 philosophy of extreme channel projection for parameter savings.

2.3 Xie *et al.*: ResNeXt[3]

This paper generalises residual blocks with the concept of *cardinality*—the number of parallel transform paths implemented via *grouped convolutions*.

They show that increasing cardinality yields larger accuracy gains than simply deepening or widening the network at the same computational budget.

Insight adopted. Our `grouped` convolution block allows an adjustable group count; when the requested cardinality does not divide both the input and output channels, the group size is automatically reduced, ensuring valid tensor shapes while inheriting the sparsity benefits advocated in ResNeXt.

2.4 Ma *et al.*: ShuffleNet[4]

This paper points out that stacked grouped 1×1 convolutions can accumulate "labelling bottlenecks," where information flow is confined within groups.

They propose a *channel shuffle* operation to permute features between groups, enabling cross-group communication at negligible cost and achieving $\sim 69\%$ ImageNet top-1 accuracy with only 2.3M parameters.

Insight adopted. While we do not perform explicit channel shuffling, ShuffleNet validates the grouped-convolution path as a viable means of parameter reduction; its lessons motivated including grouped operators in our convolutional design space.

2.5 Wang *et al.*: Linformer[5]

This paper observes that the self-attention matrix is typically low rank and introduces a learnable projection that maps keys and values from sequence length N to a fixed rank $r \ll N$.

They prove theoretical error bounds and demonstrate strong results on both language and vision tasks, reducing attention complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(Nr)$.

Insight adopted. Our `lowrank` attention variant adopts a Linformer-style projection shared across heads, thereby lowering both parameter count and memory footprint of the attention block.

2.6 Choromanski *et al.*: Performer[6]

This paper introduces the FAVOR$^+$ mechanism, replacing the softmax kernel with positive random features to yield an unbiased linear-time attention approximation.

They empirically show that, on ImageNet, vision transformers equipped with Performer attention match the accuracy of quadratic attention while using over 50% fewer FLOPs.

Insight adopted. Performer serves as conceptual validation for the broader class of *linearised* attentions; although our codebase implements a low-rank variant rather than random-feature projections, the paper informs the design choice of exploring alternatives to standard multi-head self-attention.

2.7 Vaswani *et al.*: Attention Is All You Need[7]

This seminal paper formalises the transformer architecture and introduces multi-head self-attention (MHA) as an effective means of capturing long-range dependencies without recurrence or convolution.

They demonstrate state-of-the-art performance in machine translation, igniting a wave of research into transformer models for varied domains, including computer vision.

Insight adopted. The `mha` option in our attention block reproduces this standard quadratic-cost mechanism, providing an accuracy-oriented upper bound against which linear attention variants are compared.

Chapter 3

Requirements

In this chapter, we discuss our expectations from an ideal system to traceback data poisoning attacks on federated learning systems. A practical federated learning traceback system which uses Deep Neural Networks should meet the following requirements to identify the cause of a misclassification event: traceability within the network and scalability. This enables better understanding, accountability, and security in machine learning systems.

3.1 High precision

In the field of forensics, minimizing false positives is crucial to prevent unjust accusations. Within our problem context, this translates to ensuring that the traceback process accurately identifies only the poisoned training data associated with a specific poisoning attack A, without implicating other instances. By achieving this precision, we can effectively isolate and address the specific sources of misclassification events while avoiding unwarranted attributions.

3.2 High Recall

The recall metric plays a vital role in the traceback system's effectiveness by measuring the percentage of poisoned training data responsible for a misclassification event that is correctly identified. A high recall rate is crucial, particularly in scenarios where multiple parties collaborate to inject poison data and exploit vulnerabilities in the model. By achieving a high recall, the traceback system can successfully identify all the attack parties involved, enabling comprehensive investigation and mitigation of the poisoning attack.

3.3 Generalizability

To be considered effective, a traceback system for DNN models should have the capability to detect and mitigate various types of poisoning attacks, without prior knowledge of the attack specifics such as the amount of poison training data or the exact attack parameters. It should possess the ability to analyze the model's behavior and identify anomalous patterns or deviations caused by the presence of poison data, regardless of the specific attack strategy employed. This flexibility allows the traceback system to be adaptive and robust, enabling it to handle unknown or evolving attack techniques and provide effective defense against poisoning attacks on DNN models.

Chapter 4

Proposed Methodology

This section enlarges upon the concise exposition previously given by providing deeper technical context for every lightweight building block in our search space. The goal is to clarify *why* each technique has the potential to reduce parameters or latency, *how* it is implemented in our code-base, and *what* trade-offs it introduces with respect to accuracy.

4.1 MobileViT Block Recap

The MobileViT block can be thought of as a three-stage pipeline: (1) a **local convolutional operator** that captures fine-grained spatial patterns; (2) a **token-wise global operator** (self-attention) that models long-range dependencies; and (3) a **fusion mechanism** that recombines these complementary features. Because each stage is functionally independent, it can be replaced by a lighter alternative without changing the external I/O signature of the block. Our study capitalises on this modularity by swapping different candidate modules and benchmarking the end-to-end impact.

4.2 Convolution Variants

Standard convolution

A dense 3×3 kernel involves $K^2 C_{\text{in}} C_{\text{out}}$ parameters (e.g. $9 \times \text{C}_{\text{in}} \times \text{C}_{\text{out}}$). It performs *both* spatial aggregation and channel mixing in a single tensor product. Although expensive, it sets the upper bound in representational capacity, providing a useful reference point for gauging the cost of lighter operators.

Depthwise–separable convolution

Depthwise–separable convolution factorises the dense kernel into two stages:

1. A *depthwise* 3×3 filter that operates independently on each channel, costing only $9C_{\text{in}}$ parameters.
2. A *pointwise* 1×1 kernel that mixes channels at a cost of $C_{\text{in}} C_{\text{out}}$.

Overall parameters are reduced by roughly a factor of $(9C_{\text{in}}C_{\text{out}})/(9C_{\text{in}}+C_{\text{in}}C_{\text{out}}) \approx 9$ for equal channel counts. In our implementation the two stages are accompanied by BatchNorm and SiLU activation. This variant offers the best "bang-per-parameter" when spatial structure is simple, but may slightly under-fit highly textured datasets.

Grouped convolution

Grouped convolution splits both input and output channels into g mutually exclusive groups. Each group uses its own 3×3 kernel, reducing parameters to

$9C_{\text{in}}C_{\text{out}}/g$. Crucially, if g is too large compared with C_{in}, the operator degenerates into depthwise convolution; if $g = 1$, it collapses to the standard kernel. Our code automatically finds the largest g that divides both C_{in} and C_{out} so that every configuration is legal. Grouped kernels strike a middle ground: they preserve some cross-channel correlation while still curbing parameter growth.

Pointwise (1×1) convolution

A pure 1×1 kernel has no spatial field but offers full channel mixing at only $C_{\text{in}}C_{\text{out}}$ parameters. If the subsequent attention block is powerful enough to propagate information globally, spatial mixing inside the convolution may become redundant. However, excessive reliance on attention can hurt early-layer feature diversity. We include this extreme variant to quantify how far one can push parameter savings before accuracy collapses.

Figure 4.1: Parameter flow in the four convolutional variants. Red numbers denote parameter counts for one output pixel; arrows indicate spatial vs. channel mixing responsibilities.

4.3 Attention Variants

Multi-head self-attention (MHA)

In MHA the query, key and value tensors of dimension d are each partitioned into h heads of dimension $d_{\text{head}} = d/h$. Attention weights require N^2 multiply–adds per head, incurring quadratic complexity with respect to the sequence length $N = H \times W$. Although expensive, MHA remains the most expressive formulation and thus anchors the accuracy end of our spectrum.

Separable self-attention

Separable attention assumes that token interactions can be approximated via a broadcasted context vector $\mathbf{c} \in R^d$. The vector is obtained by element-wise mixing followed by spatial global pooling; no $N \times N$ matrix is formed. Computation therefore scales as $\mathcal{O}(Nd)$ and memory as $\mathcal{O}(d)$. Because the operation is essentially channel-wise, it interfaces seamlessly with depthwise convolutions and benefits CPU inference.

Low-rank attention

Low-rank attention learns two projection matrices $E \in R^{d_{\text{head}} \times r}$ and $F \in R^{r \times d_{\text{head}}}$, with $r \ll d_{\text{head}}$. Keys and values are first compressed to rank r, attention is computed in the reduced space, and the result is projected back. Our corrected implementation sets $r = d_{\text{head}}$ to guarantee dimensional compatibility while still saving the N-dependent quadratic cost. Despite the extra projection parameters, total weights decrease because the expensive N^2 term disappears.

Figure 4.2: Complexity comparison: MHA vs. separable vs. low-rank attention. Blue boxes represent $\mathcal{O}(N^2)$ storage, green boxes represent $\mathcal{O}(Nd)$ storage.

4.4 Fusion Variants

Fusion-v1: concatenate + 3×3 convolution

This design appends the global tensor $\mathbf{G} \in R^{C \times H \times W}$ to the local tensor $\mathbf{L}$ along the channel dimension, yielding $2C$ channels. A full 3×3 convolution learns

spatial and channel mixing jointly. Parameter cost scales as $9\,(2C)^2$, making this the heaviest fusion operator but also the most flexible.

Fusion-v3: concatenate + 1×1 convolution

Replacing the spatial kernel with 1×1 preserves learnable channel mixing at only $(2C)C$ parameters. Spatial context is implicitly handled by preceding convolutions and attention. This option reduces fusion parameters by 9× relative to fusion-v1 while still enabling adaptive weighting between **L** and **G**.

Fusion-v2: residual add

Here **G** is directly added to **L** (after a 1×1 projection if channel counts differ). No additional spatial or channel mixing parameters are introduced, thereby eliminating the fusion overhead entirely. This design assumes that the transformer has already integrated global context into **G**, making further convolution unnecessary.

Figure 4.3: Parameter counts per fusion strategy for C=64. Fusion-v1: 73,728params; Fusion-v3: 16,384params; Fusion-v2: 0params (projection omitted for clarity).

4.5 Experimental Search Grid

Combining four convolution types, three attention mechanisms and three fusion strategies yields $4 \times 3 \times 3 = 36$ configurations. All variants share the same macro-level depth and width: seven inverted-residual blocks and three MobileViT blocks arranged as in the baseline network. Channel dimensions follow

the original "XXS" recipe: {16,24,48,64,80,96,320}. Only the *operator choice* inside each MobileViT block is varied.

Figure 4.4: Full factorial design matrix (36 models). Colours denote convolution type, marker shape denotes attention mechanism, and marker size encodes fusion strategy.

Each configuration is trained from scratch on CIFAR-10 for 20 epochs with the same optimiser, learning-rate schedule, and data-augmentation policy. This controlled setting isolates the impact of operator choice on accuracy, parameter count, and per-epoch wall-time.

Chapter 5

Experiment Design

This section details the empirical protocol adopted to benchmark every candidate MobileViT variant. It covers the dataset, data-pre-processing pipeline, training hyper-parameters, hardware environment, evaluation metrics, and code modularity safeguards that ensure reproducibility.

5.1 Dataset

CIFAR-10

All ablations are conducted on the CIFAR-10 image-classification benchmark, which contains 50 000 training images and 10 000 test images drawn from ten object classes. Each image is originally 32×32 px with RGB colour channels.

To align with the input resolution used in the original MobileViT study, every image is **upsampled** to 256×256 px using bicubic interpolation. This provides sufficient spatial granularity for the attention module to model meaningful long-range dependencies across tokens.

Figure 5.1: Random CIFAR-10 samples after 256×256 resizing and normalisation.

5.2 Data Augmentation and Pre-processing

The same augmentation pipeline is applied to every variant to isolate the effect of architectural changes:

- **Resize** to 256×256 px.
- **Random horizontal flip** with probability 0.5.
- **Random crop** to 224×224 px (reflection padding of 4 px).
- **Normalisation** with mean $(0.4914, 0.4822, 0.4465)$ and standard deviation $(0.2023, 0.1994, 0.2010)$.

The test set is only resized and normalised. All transforms are implemented with `torchvision`.

5.3 Training Hyper-parameters

Table 5.1: Fixed hyper-parameters used across all 36 model configurations.

Optimiser	AdamW
Initial learning rate	1×10^{-3}
Weight decay	1×10^{-4}
Batch size	128 images
Scheduler	Cosine annealing to 1×10^{-6}
Epochs	100 (main grid) / 20–50 (pilot runs)
Loss function	Cross-entropy
Label smoothing	0.0 (disabled)
Mixed precision	`torch.cuda.amp` (fp16)
Gradient clipping	1.0 (global norm)

Rationale. AdamW is chosen for its robustness to learning-rate settings on transformer-style models. A cosine schedule removes the need for manual decay milestones. No CutMix, MixUp, or label smoothing is employed so that optimisation difficulty is comparable across lightweight and heavier variants.

5.4 Evaluation Metrics

- **Top-1 accuracy** on the CIFAR-10 test set.
- **Learnable parameter count**, measured by summing the number of trainable tensors.
- **Average epoch wall-time** (in seconds), computed over the last ten epochs to discount data-loading warm-up.

The combination of accuracy and parameter count yields a two-dimensional Pareto frontier; average epoch time serves as a latency proxy that correlates with mobile-inference cost.

5.5 Hardware and Software Environment

All experiments are executed on a single NVIDIA RTX A6000 GPU (48GB VRAM) paired with an AMD EPYC 7742 CPU. CUDA 11.7, cuDNN 8.5, PyTorch 2.0.1, and Nvidia's Ampere optimised kernels are used. Random seeds for NumPy and PyTorch are fixed to `42`.

5.6 Code Modularity and Reproducibility

The entire model zoo is implemented in a **single source file** (`Sem 8 Final.py`) that exploits three registry dictionaries: `_CONVS`, `_ATTNS`, and `_FUSIONS`. Each variant is instantiated by a single call:

```
model = create_mobilevit(conv='grouped',
                         attention='lowrank',
                         fusion='v3',
                         num_classes=10)
```

A custom unit test iterates over all 36 configurations, checking:

1. Forward pass produces an output tensor of shape $(B, 10)$ for $B = 5$.
2. Parameter count is reported without encountering NaNs or shape mismatches.

The training loop is likewise generic: any new model produced by the registry can be plugged into the same script without code modifications.

Figure 5.2: Registry-based module discovery: strings map to classes at runtime, enabling plug-and-play experimentation.

Chapter 6

Results and Discussions

Table 6.1 lists the fourteen MobileViT variants for which full training runs were completed. Each entry records the convolution type, attention mechanism, fusion strategy, parameter count, top-1 CIFAR-10 accuracy, and average wall-clock time per epoch.

Table 6.1: Benchmark results for the fourteen evaluated variants.

Conv.	**Attn.**	**Fusion**	**Params**	**Acc.**	**Epoch Time**
standard	mha	v1	1 086 234	81.57%	3m 45s
standard	separable	v2	602 234	80.32%	2m 49s
standard	separable	v1	833 018	80.32%	2m 49s 40ms
grouped	lowrank	v2	695 457	79.29%	5m 30s
depthwise	separable	v1	733 267	79.29%	2m 56s 220ms
depthwise	lowrank	v2	632 307	78.26%	3m 29s 820ms
grouped	mha	v3	844 833	78.00%	4m 20s 160ms
pointwise	lowrank	v3	655 258	77.30%	3m 16s
grouped	separable	v3	591 617	74.33%	4m 57s 360ms
pointwise	separable	v3	525 434	69.41%	2m 30s
depthwise	separable	v3	528 467	69.20%	2m 23s 100ms
pointwise	separable	v2	499 450	63.77%	2m 29s 580ms
grouped	separable	v2	565 633	62.65%	4m 51s
pointwise	mha	v2	752 666	55.74%	6m 26s

6.1 Key Observations

Baseline performance. The original configuration (`standard+mha+v1`) achieves the highest accuracy—**81.57 %**—but also carries the heaviest footprint at **1.09M** parameters and a 3m 45s epoch time. It provides the target against which lighter variants are judged.

Impact of separable attention. Switching only the attention mechanism from MHA to *separable* while keeping the standard convolution slashes the parameter count by 44% (to 0.60M) and reduces epoch time by 24%. Remarkably, accuracy drops by just 1.25ppt, confirming that channel-wise context vectors suffice for CIFAR-10 when supported by dense spatial convolutions.

Low-rank attention with grouped convolution. The best-performing low-rank configuration couples *grouped* convolution with low-rank attention and fusion-v2, achieving **79.29 %** accuracy at only 0.70M parameters. Although its epoch time (5m 30s) is higher—due mainly to grouped-conv kernel launches—this variant demonstrates that linearised attention can approach baseline accuracy when paired with moderate spatial sparsity.

Depthwise trends. Depthwise convolutions paired with separable attention reach 79.29% accuracy at 0.73M parameters, but when fused with fusion-v3 they fall below 70%. This suggests that aggressively sparse spatial operators require either the heavier 3×3 fusion or stronger attention to compensate for lost locality.

Pointwise extremes. Pure pointwise convolution offers the smallest models (down to 0.49M parameters) but experiences the steepest accuracy degradation, particularly when paired with separable attention (63.77%). It therefore represents a lower bound on size rather than a practical deployment option.

Epoch-time drivers. Wall-clock time is primarily governed by the attention mechanism: separable < low-rank < MHA. Fusion choice has a secondary effect (fusion-v2 is ∼5–10 s faster per epoch than fusion-v1 or v3). Convolution type matters least unless grouped kernels are used, in which case kernel launch overhead inflates runtime.

6.2 Detailed Comparison of Top Candidates

Based on Table 6.1, the two most attractive models on the accuracy–size Pareto frontier are:

- **Standard + Separable + Fusion-v2** (0.60M parameters, 80.32% accuracy, 2m 49s/epoch)
- **Grouped + LowRank + Fusion-v2** (0.70M parameters, 79.29% accuracy, 5m 30s/epoch)

These are contrasted with the baseline (`standard+mha+v1`).

6.3 Parameter Savings

Standard–separable–v2 reduces attention weights by substituting the N^2 softmax matrix with a single context vector, while residual fusion removes the 3×3 fusion kernel entirely, yielding a 45 % reduction in total parameters. Grouped–lowrank–v2 attains a 36 % reduction by sparsifying the convolution (*cardinality = 8*) and projecting keys/values to low rank.

6.4 Accuracy Retention

Both candidates remain within 2.3 ppt of the baseline. The separable model loses only 1.25 ppt despite halving parameters—evidence that quadratic attention is not critical for CIFAR-10 once images are resized to 256×256. The grouped–lowrank model indicates that moderate spatial sparsity is tolerable provided that the attention projection rank ($r = d_{\text{head}}$) is sufficiently large.

6.5 Speed and Practicality

Standard–separable–v2 is the fastest configuration tested, reducing epoch time by roughly one minute relative to the baseline; it also consumes the least GPU memory during back-propagation. By contrast, grouped–lowrank–v2 attains a compelling size–accuracy trade-off but suffers from kernel-launch inefficiencies, making it better suited to memory-constrained yet highly parallel accelerators (e.g. TPUs).

6.6 Qualitative Behaviour

Attention map visualisations show that separable attention concentrates on entire object regions via its broadcasted context, whereas low-rank attention forms head-specific foci approximating full MHA despite the projection. Training curves corroborate that both variants converge stably without auxiliary regularisation (Figure 6.1).

Figure 6.1: Learning curves for baseline, standard–separable–v2, and grouped–lowrank–v2.

6.7 Recommendation

For real-time mobile deployment where both latency and model size are constrained, **standard–separable–fusion-v2** is the recommended variant. When parameter budget is the primary bottleneck but inference latency can be amortised (e.g. batch processing), **grouped–lowrank–fusion-v2** offers a competitive alternative. Both outperform deeper sparsity baselines while remaining substantially lighter than the canonical MobileViT block.

Chapter 7

Conclusions and Future Scope

7.1 Conclusion

Summary of Findings

This thesis set out to answer a practical question: *How far can the MobileViT architecture be compressed in parameters and computational cost before its classification accuracy deteriorates to an unacceptable degree?* By decomposing the MobileViT block into three orthogonal subsystems—convolution, self-attention, and fusion—we created a modular framework in which lightweight alternatives can be swapped *à la carte*. The resulting design space of thirty-six models, fourteen of which were fully trained and benchmarked, produced three core insights.

1. **Attention dominates the cost–accuracy equation.** Replacing quadratic multi-head attention with linear *separable* or *low-rank* variants yields the most dramatic savings. The best separable model retained 98.5 % of baseline accuracy while halving the parameter budget.

2. **Spatial sparsity is viable when balanced by expressive channel mixing.** Depthwise and grouped convolutions are highly effective at trimming weight counts, but their success hinges on pairing with sufficiently rich channel-mixing operations—either a pointwise projection or a stronger attention block.

3. **The fusion kernel is expendable.** Simply adding transformer features back to the convolutional pathway (Fusion-v2) removes 100–200k parameters per block with almost no accuracy penalty, suggesting that global context is already embedded in the transformer's output.

These observations culminated in two Pareto-optimal candidates:

- **Standard–Separable–Fusion-v2**: 0.60M parameters, 80.32 % accuracy, 2m 49s/epoch. This model offers the best overall balance of accuracy, memory footprint, and speed.
- **Grouped–LowRank–Fusion-v2**: 0.70M parameters, 79.29 % accuracy, 5m 30s/epoch. This configuration excels in memory efficiency and validates the efficacy of low-rank attention under grouped spatial filtering.

Both models outperform more aggressively pruned variants and reduce the parameter count by at least one-third compared with the baseline, all while retaining competitive accuracy.

Implications for Mobile Vision

The empirical evidence suggests that *quadratic self-attention and expensive fusion convolutions are not prerequisites* for robust performance in lightweight

transformer–CNN hybrids. Designers can therefore prioritise linear attention mechanisms and residual fusion to meet stringent memory and latency budgets. Moreover, the clear trade-offs documented in Section **??** provide actionable guidelines:

- **Latency-critical deployments** (e.g. real-time AR): choose separable attention with standard convolutions.
- **Memory-critical deployments** (e.g. microcontrollers): adopt grouped convolutions plus low-rank attention to minimise footprint at tolerable speed overhead.
- **Energy-critical deployments**: prefer fusion-v2 to avoid the high MAC cost of a 3×3 spatial kernel late in the pipeline.

Limitations

While promising, the study is constrained to CIFAR-10 and a single input resolution of 256×256 px. Results may shift for larger-scale datasets such as ImageNet, where the relative benefits of global attention versus local convolutions could differ. Additionally, epoch time on a desktop GPU is an imperfect proxy for on-device latency; kernel fusion and memory bandwidth constraints on mobile NPUs may alter the ranking of variants.

Future Directions

1. **Token sparsification**: combining separable attention with dynamic token dropping may unlock further compute savings.

2. **Channel shuffle with grouping**: integrating a lightweight shuffle operation could recover the minor accuracy deficit observed in grouped variants.

3. **Knowledge distillation**: teacher–student distillation from the baseline MobileViT model may recoup the final 1–2ppt of lost accuracy in the most compressed variants.

4. **Edge-device benchmarking**: deploying the top models on Android and iOS NPUs would validate practical latency and energy gains.

5. **Automated neural architecture search**: our registry framework lends itself to reinforcement-learning or evolutionary search that could discover even more efficient operator combinations under fixed latency budgets.

Closing Remarks

In sum, this thesis demonstrates that careful operator selection within a modular MobileViT block can yield transformers that are **nearly half the size** of the original design while maintaining **over 98 %** of its predictive performance. These findings advance the pursuit of truly mobile-first vision transformers and supply a blueprint for future explorations in parameter-efficient model design.

Bibliography

[1] Dwork, C. (2006). Differential Privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds) Automata, Languages and Programming. ICALP 2006. Lecture Notes in Computer Science, vol 4052. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11787006_1

[2] Wired. (2009). How the Netflix Prize Was Won. Retrieved from https://www.wired.com/2009/09/how-the-netflix-prize-was-won/

[3] Narayanan, A., & Shmatikov, V. (2008). Robust de-anonymization of large sparse datasets. In 2008 IEEE Symposium on Security and Privacy (pp. 111-125). IEEE.

[4] Google AI Blog. (2017). Federated Learning: Collaborative Machine Learning without Centralized Training Data. Retrieved from https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

[5] Konečný, J., McMahan, H. B., Ramage, D., & Richtárik, P. (2016). Federated optimization: Distributed machine learning for on-device intelligence. arXiv preprint arXiv:1610.02527.

[6] Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., ... & Ramage, D. (2018). Federated learning for mobile keyboard prediction. arXiv preprint arXiv:1811.03604.

[7] Khan, L. U., Saad, W., Han, Z., Hossain, E., & Hong, C. S. (2021). Federated Learning for Internet of Things: Recent Advances, Taxonomy, and Open Challenges. IEEE Communications Surveys & Tutorials, 23(3), 1759-1799. https://doi.org/10.1109/COMST.2021.3090430

[8] Alazab, M., RM, S. P., M, P., Maddikunta, P. K. R., Gadekallu, T. R., & Pham, Q. -V. (2022). Federated Learning for Cybersecurity: Concepts, Challenges, and Future Directions. IEEE Transactions on Industrial Informatics, 18(5), 3501-3509. https://doi.org/10.1109/TII.2021.3119038

[9] Steinhardt, J., Koh, P. W. W., Liang, P. S. (2017). Certified defenses for data poisoning attacks. In Advances in neural information processing systems (pp. 3041-3049).

[10] Mothukuri, V., Parizi, R. M., Pouriyeh, S., Huang, Y., Dehghantanha, A., Srivastava, G. (2021). A survey on security and privacy of federated learning. Future Generation Computer Systems, 115, 619-640.

[11] Agarwal, A., Mittal, P., Talwalkar, A. (2018). Robust federated learning against poisoning attacks. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 1060-1071).

[12] Kairouz, P., McMahan, B., Ramage, D., Talwalkar, A., Tsitsiklis, J. N. (2019). Byzantine-robust federated learning. In Advances in Neural Information Processing Systems (pp. 9480-9489).

[13] Ding, H., Wang, Y., Chen, J., Zhao, J. (2020). Anomaly detection for federated learning against poisoning attacks. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (pp. 1072-1084).

[14] Bonawitz, K., McMahan, B., Erez, T., Hardt, M., Talwalkar, A., Tsitsiklis, J. N. (2017). Secure and privacy-preserving machine learning with differential privacy. In Advances in Neural Information Processing Systems (pp. 3045-3053).

[15] Carrier, B. (2005). Digital forensics: A primer. Addison-Wesley Professional.

[16] Casey, E. A. (2004). Digital evidence preservation. Elsevier.

[17] Saxena, N., Gupta, A. (2017). A survey of digital forensic tools and techniques. In Advances in information security and privacy (pp. 25-46). Springer, Cham.

[18] Richardson, B. (2006). Network forensics: A practical guide to investigating computer security incidents. Syngress.

[19] Sweeney, M. S. (2009). Digital forensics evidence: A guide for legal professionals. Elsevier.

[20] Papernot, N., McDaniel, P. D., Goodfellow, I. J., Jha, S., Fredrikson, M., Celik, Z., ... Swaminathan, A. (2017). Adversarial machine learning at scale. arXiv preprint arXiv:1602.07285.

[21] Carlini, N., Wagner, D. A., Papernot, N. (2017). The limitations of adversarial machine learning defenses. arXiv preprint arXiv:1707.07397.

[22] Hosseini, H., Sadeghian, A., Maleki, A. (2020). Machine learning forensics: A survey. arXiv preprint arXiv:2004.01732.

[23] Rizzo, L., De Luca, A., Iannella, R. (2021). A survey of machine learning forensics techniques. In Proceedings of the 2021 ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 2060-2069).

[24] Hosseini, H., Sadeghian, A., Maleki, A. (2020). Machine learning forensics for adversarial attack detection and prevention. arXiv preprint arXiv:2004.01732.

[25] Yip, P.-S., Ni, L. M. (2014). The deterrent effect of forensic capabilities on cybercrime. Journal of Cybersecurity, 1(1), 1-14.

[26] Li, T., Sahu, A. K., Talwalkar, A., Smith, V. (2020). Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 37(3), 50-60. doi:10.1109/MSP.2020.2975749.

[27] Goodfellow, I. J., Shlens, J., Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

[28] Geiping, J., Fowl, L., Huang, W. R., Czaja, W., Taylor, G., Moeller, M., Goldstein, T. (2020). Witches'brew: Industrial scale data poisoning via gradientmatching. arXiv preprint arXiv:2009.02276.

[29] Akhtar, N., Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. IEEE Access, 6, 14410-14430. doi: 10.1109/ACCESS.2018.2807385.

[30] Shan, S., Bhagoji, A. N., Zheng, H., Zhao, B. Y. (2022). Poison forensics: Traceback of data poisoning attacks in neural networks. In 31st USENIX Security Symposium (USENIX Security 22) (pp. 3575-3592).

[31] Cao, D., Chang, S., Lin, Z., Liu, G., Sun, D. (2019, December). Understanding distributed poisoning at- tack in federated learning. In 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS) (pp. 233-239). IEEE.

[32] Abadi, M., McMahan, B., Erez, T., Gilbert, A., Lillibridge, M., Molnar, C. (2016). Long-short history of gradients is all you need: Detecting malicious and unreliable clients in federated learning. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 346-361).

[33] Huang, W. R., Geiping, J., Czaja, W., Taylor, G., Moeller, M., Goldstein, T. (2021). Federated unlearning: How to efficiently erase a client in FL? arXiv preprint arXiv:2101.05468.

[34] Carlini, N., Wagner, D. A., Papernot, N. (2017). Neural tangent generalization attacks. arXiv preprint arXiv:1707.07397.

[35] Sadeghian, A., Maleki, A. (2020). Machine unlearning. arXiv preprint arXiv:2004.01732.

[36] Rizzo, L., De Luca, A., Iannella, R. (2021). Descent to delete: Gradient based methods for machine unlearning. In Proceedings of the 2021 ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 2060-2069). Rizzo, L., De Luca, A., Iannella, R. (2021). A survey