FVNet: Harnessing Liquid Neural Dynamics for Lightweight Visual Representation

Zhenzhe Hou[1*], Xiaohui Chu[1*], Runze Hu[1], Yang Li[2], Yutao Liu[3†]

[1]School of Information and Electronics, Beijing Institute of Technology, Beijing, China
[2]Department of Electronic Engineering, Tsinghua University, Beijing, China
[3]School of Computer Science and Technology, Ocean University of China, Qingdao, China
hzzrua@126.com, 3120225380@bit.edu.cn, hrzlpk2015@gmail.com, liy.2018@tsinghua.org.cn, liuyutao2008@gmail.com

Abstract

Efficient visual backbone design remains crucial for resource-constrained computer vision applications. Inspired by the adaptive continuous-time dynamics observed in biological neurons, we propose FVNet, a novel lightweight architecture that integrates liquid neural dynamics for efficient and dynamic visual feature extraction. Central to FVNet is the Fluid Temporal Flow Unit (FTFU), which employs continuous-time equations with learnable time constants to capture spatio-temporal dependencies adaptively. By further stacking these units in a Multi-Phase Fluid Block (MPFB), our model processes features across parallel temporal scales, enabling context-aware feature encoding without incurring excessive computational overhead. Through a discrete closed-form solution, FVNet achieves the representational power of continuous-time models while avoiding the instability and overhead of iterative numerical solvers. Extensive experiments on various vision tasks demonstrate that FVNet achieves superior performance and efficiency over existing state-of-the-art lightweight networks.

Code — https://github.com/HZZRua/FVNet

Introduction

Visual network design has always been a research hotspot in the field of computer vision (He et al. 2016; Dosovitskiy et al. 2021; Liu et al. 2021; Zhang et al. 2024; Ma et al. 2024; Wang et al. 2025), exhibiting remarkable performance across vision tasks. The pursuit of lightweight and efficient visual backbone networks has become increasingly critical as computer vision applications expand to resource-constrained environments, from mobile devices to edge computing systems (Pan et al. 2022; Huang et al. 2023; Vasu et al. 2023; Ma et al. 2024). While traditional Convolutional Neural Networks (CNNs) have achieved remarkable success in computer vision tasks (Krizhevsky, Sutskever, and Hinton 2017; He et al. 2016), their static computational graphs often require substantial parameter counts and floating-point operations to capture complex visual patterns, particularly limiting their deployment in scenarios with stringent computational and memory constraints. Recent advances in neural network architectures have explored various approaches to address this efficiency-performance trade-off. Vision Transformers and their variants have demonstrated superior performance but at the cost of quadratic computational complexity with respect to input resolution (Dosovitskiy et al. 2021; Liu et al. 2022; Ge et al. 2024). Lightweight CNN architectures have made significant strides in reducing computational overhead through techniques like depthwise separable convolutions and neural architecture search (Howard et al. 2017; Tan and Le 2019a; Huang et al. 2023), yet they fundamentally operate within the paradigm of static computational graphs where the flow of information is predetermined and invariant across different inputs.

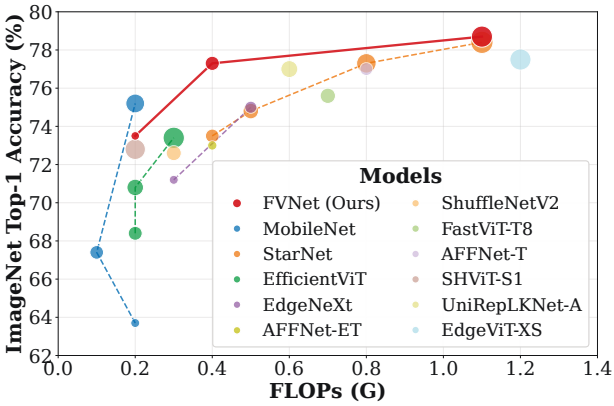

Figure 1: Mobile FLOPs VS. ImageNet Top-1 Accuracy. Bubble size indicates the model parameter number.

Liquid Neural Networks (LNNs) implement a biomimetic framework inspired by the neurophysiological mechanisms of *Caenorhabditis elegans* (Hasani et al. 2020). It presents a different computational paradigm that exhibits dynamic adaptability through continuous-time neural dynamics. Unlike conventional neural networks with fixed computational patterns, LNNs incorporate time-dependent differential equations that allow the network's behavior to modulate computational flow based on the temporal characteristics of the input and the learned time constants (Hasani et al. 2022),

*These authors contributed equally.
†Corresponding author.

leading to more efficient resource utilization and enhanced representational flexibility. This approach has demonstrated efficacy in sequential decision-making and adaptive control tasks by emulating biological neural circuit behaviors (Lechner et al. 2020; Chahine et al. 2023; Karn, Ardekani, and Abdulla 2024). The visual perception inherently involves temporal dynamics, multi-scale feature processing, and adaptive attention mechanisms that align naturally with the principles governing liquid neural networks (Lechner et al. 2020). Such an intrinsic alignment suggests that LNNs can provide more nuanced and efficient feature extraction compared to static architectures by modulating time constants and adaptive gates based on visual content complexity.

The core mechanism of LNNs is based on input-dependent synaptic time constants that dynamically reconfigure feature extraction pathways and allocate computational resources proportionate to local semantic complexity. This temporal adaptation synergizes with spatial processing through continuous-state transitions, enabling variable receptive field modulation without external gating modules (Lechner et al. 2020). Nevertheless, the inherent continuous-time computation governed by ordinary differential equations (ODEs) in liquid neural networks imposes substantial computational overhead. Recent advances in closed-form solutions for LNNs (Hasani et al. 2022) have enabled efficient discrete-time approximations, which achieve speed improvements while retaining robustness and causal reasoning capabilities. However, the practical deployment of LNNs in the field of computer vision remains largely unexplored.

To this end, we first propose a novel strategy, Fluid Temporal Flow Unit (FTFU), which aims to integrate LNNs dynamics into visual backbones, thereby achieving the LNNs' adaptive computation and exploiting intrinsic properties of continuous-time systems. Generally, our FTFU employs continuous-time liquid neural formulation for adaptive temporal processing and discrete-time closed-form solution for computational efficiency. Rather than simply applying static convolutions with fixed temporal characteristics, it firstly leverages the intrinsic dynamical properties captured by liquid neural differential equations to model the temporal relationships across feature channels. Then, parameterized by learnable time constants and amplitude scaling factors, an efficient discrete-time implementation with exponential decay mechanisms is constructed to preserve liquid dynamics while maintaining computational tractability. In this way, the continuous-time formulation captures the adaptive temporal evolution of visual features, leading to improved dynamic responsiveness akin to biological neural circuits. This design preserves liquid neural dynamics while maintaining tractability, as the continuous-time formulation captures the adaptive temporal evolution of visual features, enhancing dynamic responsiveness analogous to biological neural circuits. Thus, the derived discrete-time closed-form solution efficiently approximates the continuous dynamics without numerical integration.

In summary, the contributions of this paper are the following:

- We consider FTFU as the fundamental operation of adaptive feature processing and integrate it with other common architecture designs to form a liquid neural block MPFB.
- Building upon the liquid neural dynamics principles of MPFB, we present a new family of lightweight vision models, dubbed Fluid Vision Network (FVNet).
- Extensive experiments demonstrate that FVNet achieves superior performance and efficiency compared with existing state-of-the-art lightweight vision backbones in various vision tasks.

FVNets may serve as a baseline for integrating liquid biological neural principles into efficient vision architectures and inspire further advancements in the field of neuromorphic-inspired lightweight models.

Related Works

Liquid Neural Networks

Recent advances in LNNs have demonstrated their potential to model dynamic systems through continuous-time differential equations and closed-form approximations. Hasani et al. (Hasani et al. 2020) introduced Liquid Time-Constant (LTC) networks, leverage input-dependent synaptic gating inspired by the neurophysiology of *Caenorhabditis elegans*, enabling adaptive temporal dynamics for sequential tasks such as autonomous navigation and prediction of time series. Lechner et al. (Lechner et al. 2020) demonstrated auditable autonomy in autonomous vehicles using compact, interpretable architectures with only 19 neurons, underscoring LNNs' efficiency and transparency. Hasani et al. (Hasani et al. 2022) further improved computational efficiency by replacing iterative differential equation solvers with analytical approximations, achieving speed improvements while retaining robustness and causal reasoning capabilities. Makram et al. (Chahine et al. 2023) enabled vision-based control in unseen environments by dynamically filtering task-irrelevant features by LNNs. Karn et al. (Karn, Ardekani, and Abdulla 2024) have expanded LNNs applications beyond sequential tasks to non-causal domains, creating a unified mathematical framework that bridges temporal and spatial processing. Omran et al. (Ayoub et al. 2024) have explored how the adaptive properties of LNNs can enhance learning in dynamic environments by leveraging input-dependent time constants to mitigate catastrophic forgetting.

Lightweight Visual Backbone

The design of lightweight and efficient visual backbones has witnessed substantial evolution through innovative architectural paradigms and optimization strategies. Initial breakthroughs emerged from the development of separable convolution mechanisms and linear bottleneck network structures that achieve comparable performance with reduced computational overhead (Howard et al. 2017; Sandler et al. 2018). These foundational concepts paved the way for more sophisticated designs incorporating group-wise operations and channel manipulation techniques to enhance group information exchange (Ma et al. 2018; Chu et al. 2025b). Meanwhile, considering the limited receptive field, some

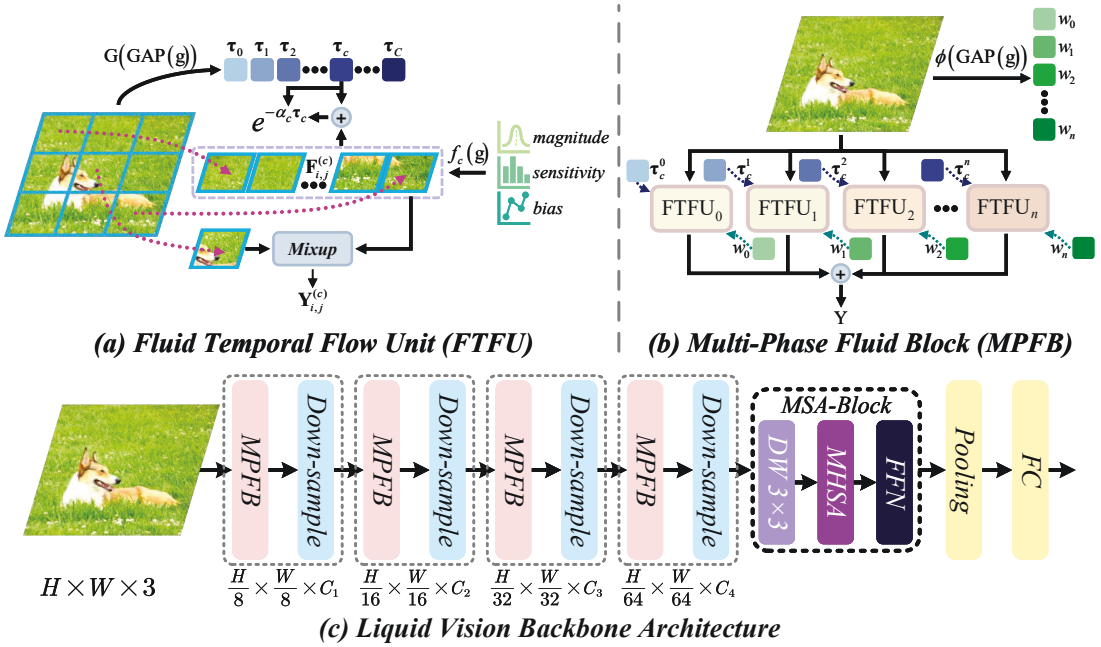

Figure 2: The overall framework of the proposed FVNet.

works have explored enhancing lightweight CNNs' capability for modeling long-range dependencies (Peng et al. 2017; Wang et al. 2025; Hu et al. 2025). The inception of Vision Transformer has catalyzed a new wave of lightweight backbone designs (Mehta and Rastegari 2023), yielding hybrid models that capitalize on both convolutional and self-attention components, demonstrated remarkable efficiency gains (Mehta and Rastegari 2022). Recent advances have explored structural innovations such as re-parameterizable components and adaptive kernel strategies to further optimize the accuracy-efficiency trade-off (Pan et al. 2022). Additionally, the development of dimension-aware design principles and frequency-domain processing techniques has opened new avenues for creating ultra-efficient visual encoders suitable for edge deployment scenarios (Li et al. 2022; Vasu et al. 2023; Chu et al. 2025a).

Methodology

Preliminaries: Liquid Neural Dynamics

We propose a new family of lightweight vision backbones for vision tasks. Our approach is grounded in the closed-form continuous-time neural network framework, where neural dynamics is characterized by first-order differential equations with state-dependent time constants. Consider a liquid neural cell with membrane potential $\boldsymbol{x}(t)$ at time t, which can be described as the solution to the following initial value problem (Hasani et al. 2020):

$$\frac{d\boldsymbol{x}(t)}{dt} = -[\boldsymbol{\tau} + f(\boldsymbol{x}(t), \mathbf{I}, \theta)] \cdot \boldsymbol{x}(t) + f(\boldsymbol{x}(t), \mathbf{I}, \theta) \cdot A, \quad (1)$$

where $\boldsymbol{\tau} \in \mathbb{R}^C$ represents the time-constant parameter vector that controls the intrinsic decay rate of neural states, the non-linear activation function $f(\boldsymbol{x}(t), \mathbf{I}, \theta)$ introduces the context-sensitive temporal modulation parameterized by θ that processes both the current state $\boldsymbol{x}(t)$ and external input $\mathbf{I}$, and A represents the reversal potential that scales the magnitude of the non-linear response.

We then reformulate Eq.(1) into a computationally stable and learnable representation by introducing learnable parameters $\alpha = \boldsymbol{\tau} + f(\boldsymbol{x}(t), \mathbf{I}, \theta)$ as the effective time constant of the system. α enables the real-time modulation of temporal response characteristics based on both internal states and external stimuli, creating a state-dependent and input-dependent temporal dynamic. Thus, the membrane potential evolution over time intervals can be expressed as:

$$\frac{d\boldsymbol{x}(t)}{dt} = -\alpha \cdot \boldsymbol{x}(t) + f(\boldsymbol{x}(t), \mathbf{I}, \theta) \cdot A. \quad (2)$$

The closed-form solution to Eq.(2) over a discrete time step Δt can be formulated as:

$$\boldsymbol{x}(t + \Delta t) = \boldsymbol{x}(t) e^{-\alpha \Delta t} + \frac{f(\boldsymbol{x}(t), \mathbf{I}, \theta) \cdot A}{\alpha} \left(1 - e^{-\alpha \Delta t}\right). \quad (3)$$

This solution avoids the instability and computational overhead of explicit numerical integration, while preserving the essential continuous-time properties (Hasani et al. 2022). The exponential decay factor $e^{-\alpha \Delta t}$ serves as a learnable factor that determines memory retention, where smaller values of α correspond to longer memory retention and larger values result in faster information decay. This relationship establishes a direct connection between the time constant parameters and the network's temporal receptive field, enabling adaptive control of the liquid neural cell attenuates memory over time. Furthermore, the state-dependent time constants provide an additional degree of freedom beyond

traditional weight-based parameterizations, expanding the representational capacity of the network (Karn, Ardekani, and Abdulla 2024).

Fluid Temporal Flow Unit

Building upon the continuous-time neural dynamics from Eq.(3), we introduce the Fluid Temporal Flow Unit (FTFU), aiming for integrating liquid neural temporal dynamics into spatial feature processing for lightweight vision models as shown in Fig.2. FTFU incorporating temporal dynamics directly into spatial feature processing, producing a unified spatio-temporal representation that adapts to input features. Specifically, FTFU operates through temporally-modulated convolution mechanism to integrate liquid time constants with spatial convolution operations. Given an input feature map $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$, the FTFU first computes a channel-wise adaptive time-constant parameter vector $\boldsymbol{\tau} \in \mathbb{R}^C$ through a learnable gate:

$$\boldsymbol{\tau} = \mathcal{G}(\text{GAP}(\mathbf{X})), \tag{4}$$

where $\mathcal{G}(\cdot)$ denotes a learnable gating function implemented as a linear transformation and $\text{GAP}(\cdot)$ represents global average pooling. The temporal computation proceeds through a discretized liquid neural dynamic where each spatial location undergoes state evolution according to Eq.(3). Let $\mathbf{X}_{i,j}^{(c)}$ denote the feature state at spatial position (i, j) and channel c, the temporal evolution of next state $\mathbf{Y}_{i,j}^{(c)}$ follows:

$$\begin{aligned}\mathbf{Y}_{i,j}^{(c)} &= \text{FTFU}(\mathbf{X}, \boldsymbol{\tau}_c) \\ &= \mathbf{X}_{i,j}^{(c)} e^{-\alpha_c \boldsymbol{\tau}_c} + f_c(\mathbf{F}_{i,j}^{(c)}) \frac{A_c}{\alpha_c}(1 - e^{-\alpha_c \boldsymbol{\tau}_c}),\end{aligned} \tag{5}$$

where $\boldsymbol{\tau}_c$ is the discrete time step of the state interval, $\mathbf{F}_{i,j}^{(c)}$ represents the local spatial neighborhood of state $\mathbf{X}_{i,j}^{(c)}$, A_c denotes the learnable reversal potential for channel c, and $\alpha_c = \boldsymbol{\tau}_c + f_c(\mathbf{F}_{i,j}^{(c)})$ constitutes the time constant incorporating both intrinsic decay and input-dependent modulation. To modulates α_c based on local feature characteristics of each state, the non-linear function $f_c(\cdot)$ is implemented as a channel-wise temporal gate with hyperbolic tangent function:

$$f_c(\mathbf{F}_{i,j}^{(c)}) = \beta_c \tanh(\gamma_c \mathbf{F}_{i,j}^{(c)} + \delta_c), \tag{6}$$

where β_c, γ_c, and δ_c are the learnable parameters that control the magnitude, sensitivity, and bias of the temporal modulation, respectively (Lechner et al. 2020). This formulation allows the FTFU to adaptively regulate its temporal dynamics based on the content of the feature map, enabling content-aware temporal integration and dynamic feature extraction of different visual patterns.

Multi-Phase Fluid Block

Using FTFU as the primary operation, we present the basic block, i.e., Multi-Phase Fluid Block (MPFB), which extends the FTFU concept to handle multi-state visual information processing through parallel liquid neural pathways. Drawing inspiration from the hierarchical processing observed in biological vision systems (Hasani et al. 2022), the MPFB is designed to process features across multiple temporal stages simultaneously.

Variant	Depths	Dimensions	Params	FLOPs
FVNet-N	[2,2,6,2]	[24,48,96,192]	1.2M	0.2G
FVNet-T	[2,3,6,3]	[32,64,128,256]	3.1M	0.4G
FVNet-S	[3,4,6,3]	[48,96,192,384]	6.8M	1.1G

Table 1: Configurations of FVNets. We only vary the block numbers and embedding dim of each stage to build different sizes of FVNet. FLOPs (G) of FVNet is measured on image crops of 224×224.

The MPFB architecture consists of parallel FTFU branches operating at different temporal stages, characterized by distinct range of time constants. Given an input feature map $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$, the network first calculates the learnable fusion weights $w_n \in \mathbb{R}^C$ for the n-th temporal state through an attention mechanism:

$$w_n = \frac{e^{\phi_n(\text{GAP}(\mathbf{X}))}}{\sum_{n=1}^{N} e^{\phi_n(\text{GAP}(\mathbf{X}))}}, \tag{7}$$

where $\phi_n(\cdot)$ denotes state-specific projection functions and N represents the number of temporal stages. This mechanism enables the network to dynamically weight and integrate information across different temporal stages based on the input features, reflecting the relative contribution of each state to the composite representation. Following the extraction of attention weights w_n, the multi-state fusion operation within the MPFB is formulated as:

$$\mathbf{Y} = \text{MPFB}(\mathbf{X}) = \sum_{n=1}^{N} w_n \cdot \text{FTFU}_n(\mathbf{X}, \boldsymbol{\tau}_c^n), \tag{8}$$

where $\mathbf{Y} \in \mathbb{R}^{H\times W\times C}$ denotes the MPFB output and $\boldsymbol{\tau}_c^n$ corresponds to the discrete time step for state n. Each state is designed to capture specific temporal dynamics at different frequencies, with smaller time steps focusing on rapid feature transitions and larger time steps capturing long-term dependencies of features.

MPFB's each pathway encodes the closed-form liquid dynamic and employs state-specific discretized time intervals. In a vision context, these parallel pathways can be interpreted as distinct "temporal lenses", each modeling the feature evolution under a different time constant. Formally, every MPFB computes its output via parallel solutions to Eq.(3) with feature-dependent modulation.

Liquid Vision Backbone Architecture

Building upon the previously discussed continuous-time liquid neural formulations, we propose a 4-stage hierarchical liquid vision architecture, Fluid Vision Network (FVNet), integrates closed-form dynamics into layered spatial processing. As illustrate in Fig.2, our FVNet is designed to capture multi-scale information flows while preserving the dynamic nature of liquid neural cells. At the initial stage, a stem layer with FTFU transforms raw image inputs into

initial liquid neural states, ensures that temporal dynamics are embedded from the earliest stage, enabling subsequent liquid neural operations to build upon established temporal states. Subsequent stages progressively refine the transformed feature through MPFB operating at different temporal scales. This hierarchical organization ensures that early stages focus on low-level feature extraction using shorter temporal spans, while deeper stages capture higher-level abstractions by adapting the time constants to longer or more specialized temporal contexts. By incrementally merging features from various scales and time constants, the backbone learns robust representations governed by the continuous-time principles established in Eq.(2) and Eq.(3).

To enable the network perform content-sensitive time integration, i.e., rapid variations in spatial inputs can trigger shorter effective steps for responsive feature updates, while more stable regions may retain memory over extended periods (Lechner et al. 2020), MPFB in each stage is designed to adaptively regulates the time-step parameter $\boldsymbol{\tau}_c^n$ in accordance with local feature statistics. Consequently, the network configures its temporal receptive field based on the evolving needs of the spatial representation, endowing it with a state-dependent adaptation that is unique to liquid neural architectures (Karn, Ardekani, and Abdulla 2024). For down-sampling, we leverage the depth-wise and point-wise convolution to reduce the spatial resolution and modulate the channel dimension, respectively. The time-constant in former stages is also propagate into next stage, enabled each deeper stage processes increasingly abstract representations, while still preserving the dynamic memory aspects of preceding layers. In the last stage, a multi-head self-attention (MHSA) operation has been incorporated to capture long-range dependencies due to the small resolution (Mehta and Rastegari 2022; Wang et al. 2025).

We bulit three FVNet variants for different computational budgets, i.e., the FVNet with nano size (FVNet-N), tiny size (FVNet-T), and small size (FVNet-S). We only vary the block numbers and embedding dim of each stage to build different sizes of FVNet, as detailed in Table 1. We present further analysis of the model's computational complexity in the Appendix, see Section 2 of the Appendix for more details.

Model	Params	FLOPs	Top-1	Latency(ms)	
	(M)	(G)	(%)	GPU	CPU
MobileNetV1-0.5	1.3	0.2	63.7	0.6	2.7
MobileViTV2-0.5	1.4	0.5	70.2	0.8	3.6
EdgeNeXt-XXS	1.3	0.3	71.2	0.7	3.3
AFFNet-ET	1.4	0.4	73.0	0.9	3.9
FVNet-N(Ours)	1.2	0.2	**73.5**	0.5	2.1
MobileNetV3-S	2.9	0.1	67.4	0.6	2.0
EfficientViT-M1	3.0	0.2	68.4	0.4	1.6
FasterNet-T0	3.9	0.3	71.9	0.8	4.3
ShuffleNetV2	3.5	0.3	72.6	0.7	3.7
StarNet-S1	2.9	0.4	73.5	0.7	3.2
StarNet-S2	3.7	0.5	74.8	0.6	3.4
EdgeNeXt-XS	2.3	0.5	75.0	1.1	6.9
FastViT-T8	3.6	0.7	75.6	1.3	6.4
AFFNet-T	2.6	0.8	77.0	1.1	5.1
FVNet-T(Ours)	3.1	0.4	**77.3**	0.8	3.1
EfficientViT-M2	4.2	0.2	70.8	0.4	1.7
SHViT-S1	6.3	0.2	72.8	0.5	1.8
EfficientViT-M3	6.9	0.3	73.4	0.6	2.2
LSNet-T	11.4	0.3	74.9	0.6	2.4
MobileNetV3-L	5.4	0.2	75.2	0.8	3.9
SHViT-S2	11.4	0.4	75.2	0.7	2.6
FasterNet-T1	7.6	0.9	76.2	1.1	7.3
UniRepLKNet-A	4.4	0.6	77.0	1.2	5.4
EfficientNet-B0	5.3	0.4	77.1	1.1	6.6
StarNet-S3	5.8	0.8	77.3	0.9	5.0
SHViT-S3	14.2	0.6	77.4	0.9	3.6
EdgeViT-XS	6.8	1.2	77.5	3.9	13.8
LSNet-S	16.1	0.5	77.8	0.7	3.1
StarNet-S4	7.5	1.1	78.4	1.2	7.1
UniRepLKNet-F	6.2	0.9	78.6	1.3	6.1
FVNet-S(Ours)	6.8	1.1	**78.7**	2.0	9.4

Table 2: Classification results on ImageNet-1K. To enable latency evaluations on different hardware, all models have been converted from Pytorch code to the ONNX format, following (Ma et al. 2024).

Experiments

Experimental Settings

Implementation details. For image classification on ImageNet-1K benchmark (Deng et al. 2009), we employ the image size of 224×224 for both training and testing, all models are trained from scratch for 300 epochs. We use the AdamW optimizer with cosine learning rate scheduler. The initial learning rate is set to 4×10^{-3}, and the total batch size is set to 2048. For data augmentation, we leverage mixup, CutMix, and random erasing, following (Huang et al. 2023; Wang et al. 2025).

For object detection and instance segmentation on CoCo-2017 benchmark (Lin et al. 2014), we employ the same training setting as (Cai et al. 2023; Vasu et al. 2023). To be specific, we integrate FVNet as the backbone model into RetinaNet (Lin et al. 2017) and Mask R-CNN (He et al. 2017), the AdamW optimizer is utilized to train these models for 12 epochs with a batch size of 16. The training resolution is 1333×800 and the initial learning rate is set to 2×10^{-4}. The learning rate decays with a rate of 0.1 at the 8-th and 11-th epochs. And for semantic segmentation on ADE20K benchmark (Zhou et al. 2017), we incorporate FVNet as the backbone model in Semantic FPN (Kirillov et al. 2019). All models are trained for 40K iterations by the AdamW optimizer with a batch size of 32. We adopt the poly-learning rate schedule with the power of 0.9 and the initial learning rate of 2×10^{-4}, following (Vasu et al. 2023; Wang et al. 2025). We employ the training resolution of 512×512 and report the single scale testing results on the ADE20K validation set. The backbone models are initial-

Backbone	FLOPs	RetinaNet						Mask R-CNN					
	(G)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
MobileNetV2	1.6	28.3	46.7	29.3	14.8	30.7	38.1	29.6	48.3	31.5	27.2	45.2	28.6
MobileNetV3	1.1	29.9	49.3	30.8	14.9	33.3	41.1	29.2	48.6	30.3	27.1	45.5	28.2
FairNAS-C	1.7	31.2	50.8	32.7	16.3	34.4	42.3	31.8	51.2	33.8	29.4	48.3	31.0
EfficientViT-M4	1.6	32.7	52.2	34.1	17.6	35.3	46.0	32.8	54.4	34.5	31.0	51.2	32.2
FVNet-N(Ours)	1.0	**33.4**	**53.1**	**35.0**	**18.2**	**36.2**	**47.1**	**33.8**	**55.5**	**35.6**	**31.9**	**52.3**	**33.3**
ResNet18	9.5	31.8	49.6	33.6	16.3	34.3	43.2	34.0	54.0	36.7	31.2	51.0	32.7
DFvT-T	6.9	-	-	-	-	-	-	34.8	56.9	37.0	32.6	53.7	34.5
StarNet-S1	2.2	33.6	53.3	35.1	18.3	36.0	47.0	33.8	56.1	35.5	31.9	52.9	33.4
LSNet-T	1.5	34.2	54.6	35.2	17.8	37.1	48.5	35.0	57.0	37.3	32.7	53.8	34.3
EfficientViT-M5	2.8	34.3	54.2	36.1	18.0	36.9	48.2	34.9	57.0	37.0	32.8	53.7	34.6
FVNet-T(Ours)	2.2	**35.3**	**55.2**	**37.0**	**19.6**	**38.1**	**49.5**	**36.0**	**58.3**	**38.3**	**33.7**	**55.1**	**35.4**
SHViT-S3	3.0	36.1	56.6	38.0	19.9	39.1	50.8	36.9	59.4	39.6	34.4	56.3	36.1
PoolFormer-S12	9.5	36.2	56.2	38.2	20.8	39.1	48.0	37.3	59.0	40.1	34.6	55.8	36.9
ResNet50	21.4	36.3	55.3	38.6	19.3	40.0	48.8	38.0	58.6	41.4	34.4	55.1	36.7
PVT-Tiny	11.8	36.7	56.9	38.9	22.6	38.8	50.0	36.7	59.2	39.3	35.1	56.7	37.3
FasterNet-S	23.8	-	-	-	-	-	-	39.9	61.2	43.6	36.9	58.1	39.7
RepViT-M1.1	7.0	-	-	-	-	-	-	39.8	61.9	43.5	37.2	58.8	40.1
FastViT-SA12	7.7	-	-	-	-	-	-	38.9	60.5	42.2	35.9	57.6	38.1
FVNet-S(Ours)	5.6	**38.7**	**59.6**	**40.4**	**22.3**	**41.5**	**53.5**	**40.1**	**62.2**	**42.6**	**37.6**	**59.1**	**39.3**

Table 3: Object detection and instance segmentation results on CoCo-2017. AP^{box} and AP^{mask} indicate bounding box AP and mask AP, respectively. Following common convention (Vasu et al. 2023; Wang et al. 2025), FLOPs (G) of backbone is measured on image crops of 512×512.

Backbone	FLOPs	mIoU	Backbone	FLOPs	mIoU
StarNet-S1	2.2	36.0	EFormer-L1	6.8	38.9
MobileNetV3	1.1	37.0	PVT-Small	23.1	39.8
PVTv2-B0	3.8	37.2	PoolFormer-S24	17.8	40.3
VAN-B0	4.5	38.5	FastViT-SA24	15.0	41.0
FVNet-N	1.0	**38.7**	StarNet-S4	4.0	41.2
FastViT-SA12	7.7	38.0	EdgeViT-XS	6.3	41.4
EdgeViT-XXS	3.2	39.7	SwiftFormer-L1	8.3	41.4
SHViT-S3	3.0	40.0	Swin-T	25.6	41.5
RepViT-M1.1	7.0	40.6	StarNet-S4	5.5	41.7
FVNet-T	2.2	**40.8**	**FVNet-S**	5.6	**42.1**

Table 4: Semantic segmentation on ADE20K. Following (Vasu et al. 2023; Wang et al. 2025), FLOPs (G) of backbone are measured on image crops of 512×512. EFormer denotes EfficientFormer.

ized with the pre-trained weights on ImageNet-1K benchmark.

For benchmark purposes, our PyTorch models are converted to the ONNX format to facilitate latency evaluations on both GPU (NVIDIA GeForce RTX 4090) and CPU (Intel Xeon Platinum 8352V CPU @ 2.10GHz), following (Ma et al. 2024).

Compared Methods. We conduct a comparative analysis between FVNet variants and existing state-of-the-art (SOTA) or classical models, namely ResNet (He et al. 2016), MobileNetV1 (Howard et al. 2017), MobileViTV2 (Sandler et al. 2018), ShuffleNetV2 (Ma et al. 2018), MobileNetV3 (Howard et al. 2019), EfficientNet (Tan and Le 2019b), FairNAS (Chu, Zhang, and Xu 2021), PVT (Wang et al. 2021), Swin-Transformer (Liu et al. 2021), EdgeNeXt (Maaz et al. 2022), EdgeViT (Pan et al. 2022), PoolFormer (Yu et al. 2022), DFvT (Gao et al. 2022), AFFNet (Huang et al. 2023), FastViT (Vasu et al. 2023), EfficientViT (Cai et al. 2023), FasterNet (Chen et al. 2023), SwiftFormer (Shaker et al. 2023), EfficientFormer (Li et al. 2023), StarNet (Ma et al. 2024), SHViT (Yun and Ro 2024), UniRepLKNet (Ding et al. 2024), RepViT (Wang et al. 2024), and LSNet (Wang et al. 2025).

Experimental Results

Image Classification. As shown in Table 2, FVNet achieves superior performance across all model sizes while maintaining competitive efficiency. Specifically, FVNet-N achieves 73.5% top-1 accuracy with 1.2M parameters and 0.2G FLOPs, FVNet-T delivers 77.3% accuracy and FVNet-S reaches 78.7% top-1 accuracy, establishing new SOTA results among lightweight models while maintaining reasonable computational cost.

Object Detection. Table 3 demonstrates FVNet's strong performance across different detection architectures. With RetinaNet, FVNet-N achieves 33.4 AP, outperforming

Model	FLOPs	IN-C (↓)	IN-A	IN-R	IN-S
MobileNetV1-0.5	0.2	96.8	1.8	26.1	14.2
EdgeNeXt-XXS	0.3	94.6	3.6	29.5	18.5
EfficientViT-M1	0.2	88.5	2.7	29.4	17.8
MobileViTV2-0.5	0.5	82.3	3.4	31.8	19.5
AFFNet-ET	0.4	79.1	4.2	33.2	21.3
StarNet-S1	0.4	77.5	4.5	34.1	21.8
FVNet-N(Ours)	0.2	**75.8**	**5.1**	**35.7**	**23.2**
FasterNet-T0	0.3	89.8	2.3	28.6	16.3
ShuffleNetV2	0.3	83.7	3.8	30.9	19.2
EfficientViT-M2	0.2	78.9	4.1	32.5	20.1
StarNet-S2	0.5	73.2	5.8	37.4	25.1
FVNet-T(Ours)	0.4	**71.5**	**6.9**	**38.8**	**26.4**
EfficientViT-M3	0.3	71.1	5.2	36.1	23.4
StarNet-S3	0.8	69.3	7.8	39.2	27.8
LSNet-T	0.3	68.2	6.7	38.5	25.5
UniRepLKNet-A	0.6	67.0	8.4	37.9	26.0
SHViT-S3	0.6	66.8	9.1	38.6	26.9
FastViT-T12	1.4	64.3	14.0	39.9	27.6
RepViT-M1.1	1.3	63.7	14.8	40.7	28.3
FVNet-S(Ours)	1.1	**62.4**	**15.2**	**41.3**	**29.8**

Table 5: Robustness evaluation results on benchmark datasets. We report mCE for ImageNet-C (IN-C) and Top-1 accuracies (%) for ImageNet-A (IN-A), ImageNet-R (IN-R), and ImageNet-Sketch (IN-S). ↓ represents lower is better.

EfficientViT-M4 with lower computational cost. FVNet-T delivers 35.3 AP, surpassing LSNet-T by 1.1 AP while maintaining similar efficiency. For Mask R-CNN, FVNet-S achieves 40.1 AP^{box} and 37.6 AP^{mask}, establishing new SOTA results among efficient backbones.

Semantic Segmentation. Table 4 shows that FVNet-N achieves 38.7 mIoU with 1.0G FLOPs, outperforming VAN-B0 by 0.2 mIoU while requiring significantly fewer computations. FVNet-T delivers 40.8 mIoU, surpassing RepViT-M1.1 by 0.2 mIoU with much lower computational cost. FVNet-S achieves the best performance at 42.1 mIoU.

Robustness Evaluation. We conduct robustness evaluation for FVNet on various benchmarks, including ImageNet-C (Hendrycks and Dietterich 2019), ImageNet-A (Hendrycks et al. 2021b), ImageNet-R (Hendrycks et al. 2021a), and ImageNet-Sketch (Wang et al. 2019). Following (Vasu et al. 2023; Wang et al. 2025), we report mean corruption error (mCE) for ImageNet-C and top-1 accuracies for other datasets. As shown in Table 5, FVNet shows strong domain generalization capabilities and promising robustness to corruptions, achieving SOTA performance.

FTFU Effectiveness			Architecture Components		
Model	FLOPs	Top-1	Model	FLOPs	Top-1
DWConv(2017)	0.4	76.1	No Stage	0.4	75.8
DY-Conv(2020)	0.4	76.3	+Stage1	0.4	76.2
Invo.(2021)	0.4	76.5	+Stage2	0.4	76.8
MBConv(2019b)	0.4	76.8	+Stage3	0.4	77.1
PConv(2023)	0.4	76.4	w/o MHSA	0.4	77.1
LSConv(2025)	0.4	77.0	w/o GAP	0.4	76.9
FTFU	0.4	**77.3**	**FVNet-T**	0.4	**77.3**

Table 6: Ablation study on FTFU effectiveness and architecture components.

Fusion Strategies in MPFB			Activation Functions		
Strategy	FLOPs	Top-1	Function	FLOPs	Top-1
Average	0.4	76.4	Linear	0.4	76.1
Concat	0.4	76.8	ReLU	0.4	76.5
SE-based	0.4	77.0	Sigmoid	0.4	76.9
Attention	0.4	**77.3**	**Tanh**	0.4	**77.3**

Table 7: Ablation study on fusion strategies in MPFB and activation functions.

Ablation Studies

We conduct comprehensive ablation studies to validate the effectiveness of key components in FVNet. All experiments are performed on ImageNet-1K using FVNet-T. Table 6 presents our analysis on FTFU effectiveness and architecture components. For FTFU effectiveness, we replace the temporal evolution mechanism with various convolution operations while maintaining identical architectures. FTFU achieves 1.5% improvement over DWConv and outperforms recent efficient blocks including MBConv and LSConv. For architecture design study, we progressively add MPFB to different stages to analyze their individual contributions. The results show progressive improvements with each stage, validating the hierarchical integration of temporal dynamics. Removing MHSA or GAP components results in performance degradation, confirming their necessity. Table 7 examines fusion strategies in MPFB and activation functions in Eq.6. For temporal dynamics, we compare different methods for combining multi-phase temporal features. Attention-based fusion achieves superior performance over simpler alternatives like average pooling and squeeze-and-excitation-based (SE-based) fusion. The activation functions study shows that Tanh activation ensures stable temporal dynamics, outperforming linear, ReLU, and Sigmoid alternatives.

Conclusion

In this paper, we presented FVNet, a novel lightweight vision architecture that integrates liquid neural dynamics for efficient visual feature extraction. Our approach introduces the Fluid Temporal Flow Unit (FTFU), which employs continuous-time equations with learnable time constants to capture adaptive spatio-temporal dependencies, and the Multi-Phase Fluid Block (MPFB) for parallel multi-scale temporal processing. Extensive experiments across multiple vision tasks validate FVNet's effectiveness. In the future, we will focus on exploring more sophisticated temporal dynamics formulations and investigating the potential of liquid neural principles in other computer vision domains.

Acknowledgments

This work was supported in part by the National Science Foundation of China under Grant 62201538 and Grant 62301041, and in part by the Shandong Natural Science Foundation under Grant ZR2022QF006 and Grant ZR2024MF116.

References

Ayoub, O.; Andreoletti, D.; Knapińska, A.; Goścień, R.; Lechowicz, P.; Leidi, T.; Giordano, S.; Rottondi, C.; and Walkowiak, K. 2024. Liquid Neural Network-based Adaptive Learning vs. Incremental Learning for Link Load Prediction amid Concept Drift due to Network Failures. *arXiv preprint arXiv:2404.05304*.

Cai, H.; Li, J.; Hu, M.; Gan, C.; and Han, S. 2023. Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 17302–17313.

Chahine, M.; Hasani, R.; Kao, P.; Ray, A.; Shubert, R.; Lechner, M.; Amini, A.; and Rus, D. 2023. Robust flight navigation out of distribution with liquid neural networks. *Science Robotics*, 8(77): eadc8892.

Chen, J.; Kao, S.-h.; He, H.; Zhuo, W.; Wen, S.; Lee, C.-H.; and Chan, S.-H. G. 2023. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12021–12031.

Chen, Y.; Dai, X.; Liu, M.; Chen, D.; Yuan, L.; and Liu, Z. 2020. Dynamic Convolution: Attention Over Convolution Kernels. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11027–11036.

Chu, X.; Duan, H.; Wen, Z.; Xu, L.; Hu, R.; and Xiang, W. 2025a. Union-Domain Knowledge Distillation for Underwater Acoustic Target Recognition. *IEEE Transactions on Geoscience and Remote Sensing*, 63: 1–16.

Chu, X.; Zhang, B.; and Xu, R. 2021. FairNAS: Rethinking Evaluation Fairness of Weight Sharing Neural Architecture Search. In *International Conference on Computer Vision*.

Chu, X.; Zhou, H.; Zhang, Y.; Zhang, Y.; Hu, R.; Duan, H.; Huang, Y.; Zheng, Y.; and Ji, R. 2025b. Attention-driven acoustic properties learning for underwater target ranging. *Pattern Recognition*, 164: 111560.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Ding, X.; Zhang, Y.; Ge, Y.; Zhao, S.; Song, L.; Yue, X.; and Shan, Y. 2024. UniRepLKNet: A Universal Perception Large-Kernel ConvNet for Audio Video Point Cloud Time-Series and Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5513–5524.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Gao, L.; Nie, D.; Li, B.; and Ren, X. 2022. Doubly-Fused ViT: Fuse Information from Vision Transformer Doubly with Local Representation. In *Computer Vision – ECCV 2022*, 744–761. Cham: Springer Nature Switzerland. ISBN 978-3-031-20050-2.

Ge, C.; Ding, X.; Tong, Z.; Yuan, L.; Wang, J.; Song, Y.; and Luo, P. 2024. Advancing Vision Transformers with Group-Mix Attention. *arXiv preprint arXiv:2311.15157*.

Hasani, R.; Lechner, M.; Amini, A.; Liebenwein, L.; Ray, A.; Tschaikowski, M.; Teschl, G.; and Rus, D. 2022. Closed-form continuous-time neural networks. *Nature Machine Intelligence*.

Hasani, R.; Lechner, M.; Amini, A.; Rus, D.; and Grosu, R. 2020. Liquid Time-constant Networks. In *AAAI Conference on Artificial Intelligence*.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2980–2988.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hendrycks, D.; Basart, S.; Mu, N.; Kadavath, S.; Wang, F.; Dorundo, E.; Desai, R.; Zhu, T.; Parajuli, S.; Guo, M.; Song, D.; Steinhardt, J.; and Gilmer, J. 2021a. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 8340–8349.

Hendrycks, D.; and Dietterich, T. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. *Proceedings of the International Conference on Learning Representations*.

Hendrycks, D.; Zhao, K.; Basart, S.; Steinhardt, J.; and Song, D. 2021b. Natural Adversarial Examples. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15257–15266.

Howard, A.; Sandler, M.; Chen, B.; Wang, W.; Chen, L.-C.; Tan, M.; Chu, G.; Vasudevan, V.; Zhu, Y.; Pang, R.; Adam, H.; and Le, Q. 2019. Searching for MobileNetV3. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 1314–1324.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861.

Hu, R.; Chu, X.; Dou, D.; Liu, X.; Liu, Y.; and Qi, B. 2025. Toward Real-World Applicability: Lightweight Underwater Acoustic Localization Model Through Knowledge Distillation. *IEEE Journal of Oceanic Engineering*, 50(2): 1429–1442.

Huang, Z.; Zhang, Z.; Lan, C.; Zha, Z.-J.; Lu, Y.; and Guo, B. 2023. Adaptive Frequency Filters As Efficient Global Token Mixers. In *ICCV*.

Karn, P. K.; Ardekani, I.; and Abdulla, W. H. 2024. Generalized Framework for Liquid Neural Network upon Sequential and Non-Sequential Tasks. *Mathematics*, 12(16).

Kirillov, A.; Girshick, R.; He, K.; and Dollár, P. 2019. Panoptic Feature Pyramid Networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6392–6401.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2017. ImageNet classification with deep convolutional neural networks. *Commun. ACM*, 60(6): 84–90.

Lechner, M.; Hasani, R.; Amini, A.; Henzinger, T. A.; Rus, D.; and Grosu, R. 2020. Neural circuit policies enabling auditable autonomy. *Nature Machine Intelligence*, 2(10): 642–652.

Li, D.; Hu, J.; Wang, C.; Li, X.; She, Q.; Zhu, L.; Zhang, T.; and Chen, Q. 2021. Involution: Inverting the Inherence of Convolution for Visual Recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, Y.; Hu, J.; Wen, Y.; Evangelidis, G.; Salahi, K.; Wang, Y.; Tulyakov, S.; and Ren, J. 2023. Rethinking Vision Transformers for MobileNet Size and Speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 16889–16900.

Li, Y.; Yuan, G.; Wen, Y.; Hu, J.; Evangelidis, G.; Tulyakov, S.; Wang, Y.; and Ren, J. 2022. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 35: 12934–12949.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal Loss for Dense Object Detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2999–3007.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*, 740–755. Cham: Springer International Publishing.

Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; Wei, F.; and Guo, B. 2022. Swin Transformer V2: Scaling Up Capacity and Resolution. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 10012–10022.

Ma, N.; Zhang, X.; Zheng, H.-T.; and Sun, J. 2018. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., eds., *Computer Vision – ECCV 2018*, 122–138. Cham: Springer International Publishing.

Ma, X.; Dai, X.; Bai, Y.; Wang, Y.; and Fu, Y. 2024. Rewrite the Stars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Maaz, M.; Shaker, A.; Cholakkal, H.; Khan, S.; Zamir, S. W.; Anwer, R. M.; and Khan, F. S. 2022. EdgeNeXt: Efficiently Amalgamated CNN-Transformer Architecture for Mobile Vision Applications. In *International Workshop on Computational Aspects of Deep Learning at 17th European Conference on Computer Vision (CADL2022)*. Springer.

Mehta, S.; and Rastegari, M. 2022. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. In *ICLR*.

Mehta, S.; and Rastegari, M. 2023. Separable Self-attention for Mobile Vision Transformers. *Trans. Mach. Learn. Res.*, 2023.

Pan, J.; Bulat, A.; Tan, F.; Zhu, X.; Dudziak, L.; Li, H.; Tzimiropoulos, G.; and Martinez, B. 2022. EdgeViTs: Competing Light-weight CNNs on Mobile Devices with Vision Transformers. In *European Conference on Computer Vision*.

Peng, C.; Zhang, X.; Yu, G.; Luo, G.; and Sun, J. 2017. Large Kernel Matters — Improve Semantic Segmentation by Global Convolutional Network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1743–1751.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4510–4520.

Shaker, A.; Maaz, M.; Rasheed, H.; Khan, S.; Yang, M.-H.; and Khan, F. S. 2023. SwiftFormer: Efficient Additive Attention for Transformer-based Real-time Mobile Vision Applications. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Tan, M.; and Le, Q. 2019a. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 6105–6114. PMLR.

Tan, M.; and Le, Q. 2019b. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, 6105–6114. PMLR.

Vasu, P. K. A.; Gabriel, J.; Zhu, J.; Tuzel, O.; and Ranjan, A. 2023. FastViT: A Fast Hybrid Vision Transformer using Structural Reparameterization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Wang, A.; Chen, H.; Lin, Z.; Han, J.; and Ding, G. 2024. Rep ViT: Revisiting Mobile CNN From ViT Perspective. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15909–15920.

Wang, A.; Chen, H.; Lin, Z.; Han, J.; and Ding, G. 2025. LSNet: See Large, Focus Small. arXiv:2503.23135.

Wang, H.; Ge, S.; Lipton, Z.; and Xing, E. P. 2019. Learning Robust Global Representations by Penalizing Local Predictive Power. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 548–558.

Yu, W.; Luo, M.; Zhou, P.; Si, C.; Zhou, Y.; Wang, X.; Feng, J.; and Yan, S. 2022. MetaFormer is Actually What You Need for Vision. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10809–10819.

Yun, S.; and Ro, Y. 2024. SHViT: Single-Head Vision Transformer with Memory Efficient Macro Design. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5756–5767.

Zhang, Q.; Zhang, J.; Xu, Y.; and Tao, D. 2024. Vision Transformer with Quadrangle Attention. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; and Torralba, A. 2017. Scene Parsing through ADE20K Dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5122–5130.