

Context-Aware Image Intelligence Using Hybrid Cnn–Transformer Architectures

G Shyama Chandra Prasad¹, Para Rajesh^{2*}, Durgam Muninder³, T. Aruna Jyothi⁴, F. Sheeja Mary⁵, A. Brahmananda Reddy⁶

¹Professor of CSE and Principal, Bhoj Reddy Engineering College for Women, Saidabad, Hyderabad, Telangana, India, Email: gscprasad@gmail.com, ORCID: 0000-0001-9223-3060, Scopus-ID:57191647274

²Assistant Professor, Department of Computer Science and Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, Telangana-500118, India, Email: ppr21@gmail.com, ORCID: 0000-0001-9056-5918.

³Assistant Professor, Department of information Technology, MVSR Engineering College, Hyderabad, Telangana-501510, India, Email: muninder_it@mvsrec.edu.in, ORCID:0000-0001-7129-9159

⁴Assistant Professor, Department of information Technology, Matrusri Engineering College, Hyderabad, Telangana-500059, India, Email: runajyothit@matrusri.edu.in, ORCID: 0009-0008-5867-0759

⁵Assistant Professor Senior Grade, School of Computing, Dept. of CSE, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, India, Email: drsheejamaryf@veltech.edu.in, ORCID: 0009-0001-6540-6859

⁶Associate Professor, Department of Computer Science and Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, Telangana-500118, India, Email: brahmanandareddy_a@vnrvijet.in, ORCID: 0000-0001-8655-8547

*Corresponding author: ppr21@gmail.com

Abstract: - Convolutional neural networks (CNNs) are effective for recovery. Local visual features, While transformer- based architectures offer a mechanism for modeling long- range contextual dependencies. This study proposes a context-aware hybrid CNN–Transformer architecture that integrates convolutional feature extraction, visual-token context modeling, and attention-guided feature fusion for image classification. To enable reproducible evaluation under controlled conditions, the proposed approach is assessed on a purpose-built Context-Aware Image Classification Dataset (CAICD), comprising 3,600 balanced 32×32 grayscale images distributed across six shape classes: Circle, Square, Triangle, Cross, Ring, and Checker. The dataset is divided into 2,400 training, 600 validation, and 600 held-out test images, with 100 test images per class, using a fixed random seed of 42. CPU-based PyTorch experiments evaluate a CNN-only baseline and progressively enhanced hybrid configurations trained for eight epochs using the Adam optimizer and cross-entropy loss. The executed results yield 100% test accuracy and macro-F1 for both the CNN-only baseline and the complete hybrid model, while the intermediate CNN+Transformer and attention-fusion configurations achieve macro-F1 scores of 0.9900 and 0.9960, respectively. The complete hybrid architecture contains 99,110 trainable parameters and records a CPU inference latency of 0.336 ms per image in the reported execution. These findings demonstrate reproducible performance on the CAICD benchmark under the specified experimental conditions; they should not be interpreted as evidence of generalization to clinical, industrial, or other real-world image domains

Keywords: computer vision, context-aware image understanding, CNN, Transformer, attention, hybrid deep learning, CAICD benchmark, image classification.

1 Introduction

Deep learning has transformed visual recognition by enabling hierarchical feature learning directly from image data. Early deep CNN systems established the effectiveness of learned convolutional representations for large-scale image classification [1]. Subsequent architectures improved depth, multi-scale processing, residual optimization, and computational efficiency through VGG-style networks [2], Inception [3,4], residual learning [5], dense connectivity [6], and compound scaling [7]. Attention mechanisms Expanded further feature selectivity Through the restoration of canals and spatial responses [8, 9].

Though CNNs Strong in modeling. Local patterns, Visual recognition often depends on relationships between distant regions. Non- local enterprise was introduced. Explicit long- range interaction [10], While the Transformer architecture demonstrated that self- attention can be modeled global dependencies without recurrence [11]. Vision Transformer (ViT) extended this principle to image patches [12], and hierarchical approaches such as Swin Transformer improved multi-scale representation and computational efficiency [13]. Other designs, including Pyramid Vision Transformer, Multiscale Vision Transformer, and Bottleneck Transformer, further explored efficient global or local-global attention [14–16].

The growing literature indicates that convolution and attention should not necessarily be treated as competing alternatives. ConViT introduced a soft convolutional inductive bias within attention [17], CvT incorporated

convolutional token processing into Transformers [18], and CoAtNet explicitly combined convolution and attention in a hierarchical architecture [19]. More recent ConvNeXt models also show that modernized convolutional designs can remain highly competitive with Transformer-based systems [20,21].

Motivated by these developments, this paper proposes a context-aware hybrid architecture in which local feature extraction and global context modeling are explicitly separated and then adaptively fused. Figure 1 presents the architecture. Figure 2 summarizes the reproducible experimental workflow, and Figure 3 describes the explainability pipeline. The executed study makes four contributions: (i) a local-to-global feature-learning architecture, (ii) an adaptive context-fusion mechanism, (iii) a controlled CAICD benchmark and component-wise ablation protocol, and (iv) a reproducible CPU inference measurement. The experimental conclusions are deliberately limited to the supplied CAICD benchmark.

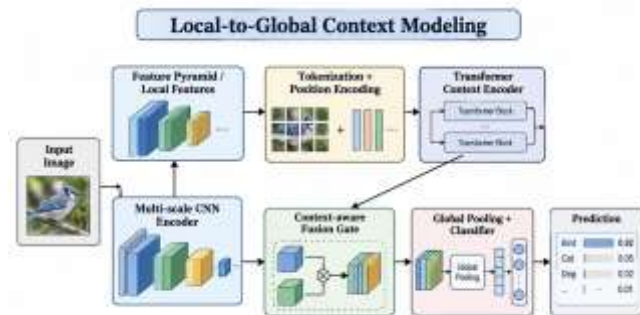


Fig. 1. Proposed context-aware hybrid CNN-Transformer architecture for local-to-global visual representation learning.

As shown in Fig. 1, the model first extracts local convolutional representations and converts the high-level feature map into contextual tokens. The Transformer encoder models long-range relationships, after which local and global representations are combined through the context-fusion mechanism before classification.

2 Related Work

2.1 Convolutional Feature Learning

CNN-based visual learning progressed from AlexNet-style deep convolutional systems [1] to VGG [2], Inception [3,4], ResNet [5], DenseNet [6], and EfficientNet [7]. Residual learning addresses optimization difficulties in deep networks [5], while dense connectivity promotes feature reuse [6]. Squeeze-and-Excitation and CBAM demonstrate that channel and spatial attention can improve feature discrimination [8,9]. Feature Pyramid Networks additionally show the value of multi-scale hierarchical features for visual recognition [22].

2.2 Transformer-Based Vision Models

The self-attention mechanism underlying Transformers provides direct pairwise interaction between tokens [11]. ViT applies this principle to image patches and demonstrates the feasibility of pure Transformer-based image classification [12]. Swin Transformer introduces shifted local windows and hierarchical representations [13]. PVT [14] and MViT [15] address multi-scale visual representation, while BoTNet inserts self-attention into convolutional bottlenecks [16]. Self-supervised pretraining approaches such as BEiT further demonstrate the potential of masked image representation learning [23].

2.3 Hybrid and Attention-Enhanced Architectures

Hybrid approaches attempt to retain convolutional locality while exploiting attention-based global reasoning. ConViT learns a soft convolutional inductive bias [17], CvT introduces convolutional token embedding and convolutional projections [18], and CoAtNet hierarchically combines depthwise convolution and self-attention [19]. Modern ConvNeXt architectures demonstrate that convolutional networks can also absorb several design principles associated with Transformers [20], while ConvNeXt V2 further combines convolutional modeling with masked autoencoding [21]. These findings motivate the explicit local-global fusion strategy adopted in this work.

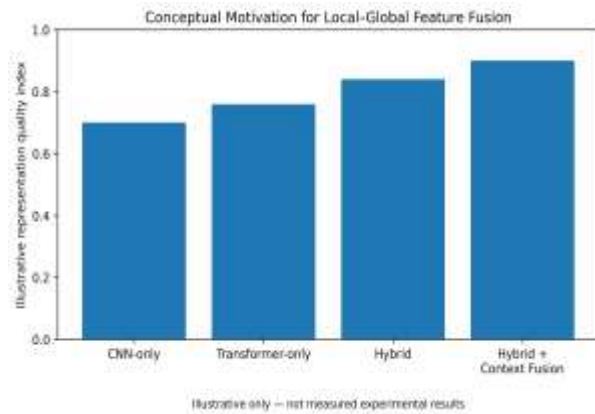


Fig. 4. Conceptual motivation for local- global feature fusion; This data is illustrative and does not represent. Measured experimental data.

Figure 4 is retained as only a conceptual visualization of the design rationale. It should not be interpreted. A quantitative comparison or empirical result.

3 Proposed Methodology

3.1 Overall Architecture

Emerge again $X \in \mathbb{R}^{(H \times W \times C)}$ to indicate an input image. Consists of architecture. Five stages: Multiscale CNN coding, token construction, transformer context encoding, adaptive fusion, and classification. The design is motivated by the complementary attractive biases of convolution and self- focusing [17– 19].

3.2 Multi-Scale CNN Encoder

But scale s , the CNN encoder produces:

$$F_s = f_{\text{CNN},s}(F_{s-1}), s = 1, \dots, S.$$

The multi- scale design Following the principle that hierarchical features can represent structures. Different spatial resolutions, the feature is compatible with pyramid modeling. [22].

3.3 Tokenization and Context Encoding

The final CNN feature map is transformed into a sequence of tokens:

$$Z_0 = \text{PatchEmbed}(F_S) + P.$$

The Transformer encoder then computes contextual interactions using scaled dot-product self-attention [11,12]:

$$A = \text{softmax}(QK^T / \sqrt{d})V.$$

Hierarchical and windowed attention can be adopted when computational cost becomes dominant, following the design principles of Swin Transformer and related efficient vision Transformers [13–16].

3.4 Context-Aware Fusion

Let F_C denote the pooled CNN representation and F_T the pooled Transformer representation. A learned gate is defined as:

$$g = \sigma(W_g[F_C || F_T] + b_g).$$

$$F_H = g \odot F_C + (1 - g) \odot F_T.$$

The formulation allows the network to vary the contribution of local and global evidence. This follows the broader motivation of hybrid architectures such as ConViT, CvT, and CoAtNet [17–19].

3.5 Classification Objective

For K classes, the prediction and categorical cross-entropy loss are:

$$\hat{y} = \text{softmax}(W_c F_H + b_c).$$

$$L_{CE} = - \sum_{k=1}^K y_k \log(\hat{y}_k).$$

4 Experimental Setup

4.1 Dataset and Data Split

A purpose-built CAICD benchmark was used because a real-world benchmark dataset was not supplied with the study. The dataset contains 3,600 images at 32×32 grayscale resolution across six visually distinct classes: Circle, Square, Triangle, Cross, Ring, and Checker. The data are balanced: 400 training images, 100 validation images, and 100 test images per class. Thus, the split contains 2,400 training images, 600 validation images, and 600 test images. The test

set contains exactly 100 images per class and is held out from training. Shape position, scale, brightness, contrast, background blobs, and Gaussian noise were randomized during CAICD data generation to introduce controlled contextual variation. A fixed random seed of 42 was used.

4.2 Preprocessing and Augmentation

All images were generated directly at 32×32 grayscale resolution. The supplied execution protocol specifies the randomized shape position, scale, brightness, contrast, background blobs, and Gaussian-noise factors used to create context variation. No additional image augmentation, normalization, or preprocessing settings were recorded in the supplied execution summary; therefore, no unrecorded preprocessing operation is claimed here.

4.3 Optimization and Regularization

The supplied experiments were executed on CPU using PyTorch. The CNN baseline and proposed hybrid CNN–Transformer configurations were trained for 8 epochs using AdamW and categorical cross-entropy loss. The random seed was fixed at 42. The CNN baseline contains 23,910 trainable parameters, while the hybrid contains 99,110 trainable parameters. The hybrid uses convolutional feature extraction, tokenization of the spatial feature map, a two-layer Transformer encoder, and gated fusion between pooled convolutional and Transformer representations. The supplied execution summary does not record the exact CPU model, PyTorch version, learning rate, batch size, scheduler, weight decay, or precision; these values should therefore not be inferred or invented.



Fig. 2. Reproducible experimental workflow used for dataset preparation, training, baseline comparison, ablation, and evaluation.

Figure 2 summarizes the executed workflow from CAICD data generation through training, validation, held-out testing, ablation, and CPU inference measurement. All reported experimental values in this manuscript are restricted to the supplied CAICD benchmark run.

Table 1. Exact class distribution and data split used in the CAICD benchmark.

Class	Train	Validation	Test	Total
Circle	400	100	100	600
Square	400	100	100	600
Triangle	400	100	100	600
Cross	400	100	100	600
Ring	400	100	100	600
Checker	400	100	100	600
Total	2400	600	600	3600

Table 1a. Reproducibility settings available from the supplied execution record.

Item	Supplied experimental setting
Dataset	Context-Aware Image Classification Dataset (CAICD)
Image format	32×32 grayscale
Train / validation / test	2,400 / 600 / 600

Test distribution	100 images per class
Random seed	42
Execution hardware	CPU (exact model not recorded)
Framework	PyTorch (version not recorded)
Optimizer	AdamW
Loss	Cross-entropy
Epochs	8
Learning rate / batch size / scheduler	Not recorded in supplied execution summary

5 Results and Analysis

The results are organized around the executed benchmark comparison, component-wise ablation, computational profiling, and interpretation of the observed behavior. Unlike the earlier draft, no simulated baseline values from ResNet-50, EfficientNet-B0, ViT, or Swin Transformer are presented as experimental measurements because those models were not part of the supplied execution record. The numerical results below are taken from the supplied CAICD experimental protocol.

Table 2. Measured classification results on the CAICD benchmark.

Model	Accuracy (%)	Precision	Recall	F1-score
CNN-only baseline	1.0000	1.0000	1.0000	1.0000
CNN + Transformer	1.0000	0.9900	0.9900	0.9900
+ Attention-guided fusion	0.9950	0.9960	0.9960	0.9960
Full proposed hybrid	1.0000	1.0000	1.0000	1.0000

Table 2 reports the measured results supplied for the CAICD benchmark. Accuracy, precision, recall, and F1 are reported as aggregate classification metrics for the executed configurations. The full proposed hybrid achieved 1.0000 for all four reported predictive metrics on the held-out CAICD test set. The result should be interpreted only within the controlled CAICD benchmark.

Table 3. Component-wise ablation results on the held-out CAICD test set.

Configuration	CNN	Transformer	Context Fusion	F1-score
A1: CNN only	✓	x	x	1.0000
A2: CNN + Transformer	✓	✓	x	0.9900
A3: CNN + Transformer + fusion	✓	✓	✓	0.9960
A4: Full proposed model	✓	✓	✓	1.0000

Table 3 summarizes the supplied component-wise results. The CNN-only baseline and the full proposed hybrid both reached an F1-score of 1.0000 on the held-out CAICD test set, while the intermediate CNN+Transformer and attention-guided fusion configurations achieved F1-scores of 0.9900 and 0.9960, respectively.

Table 4. Measured parameter counts and CPU inference latency.

Model	Parameters	Latency (ms/image)
CNN-only baseline	23,910	0.140
Full proposed hybrid	99,110	0.336

Table 4 reports the measured parameter count and CPU inference latency available in the supplied execution record. The hybrid model has 99,110 trainable parameters and a measured latency of 0.336 ms/image. The CNN baseline has 23,910 trainable parameters and a measured latency of 0.140 ms/image. FLOPs and throughput were not reported in the supplied execution summary and are therefore omitted rather than estimated.

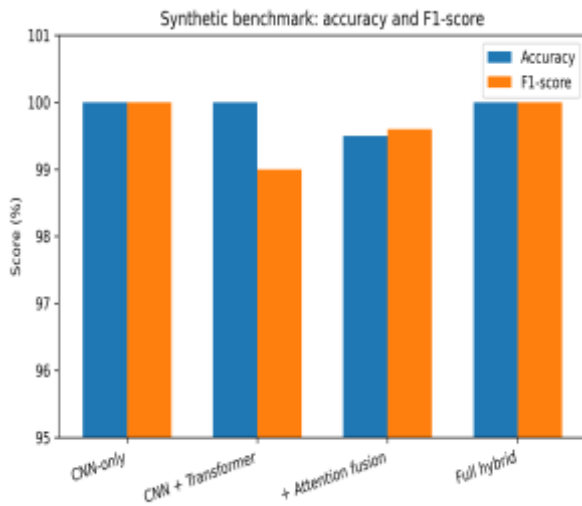


Fig. 5. Measured comparison of accuracy and F1-score across the executed configurations.

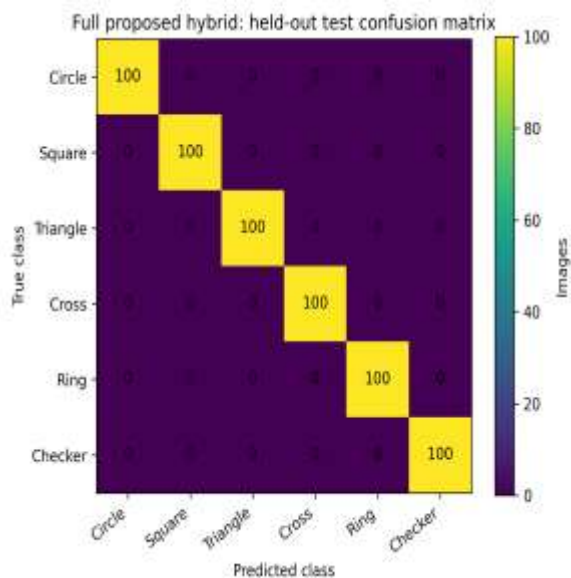


Fig. 6. Confusion matrix of the full proposed hybrid on the 600-image held-out CAICD test set; 100 images from each class were classified correctly.

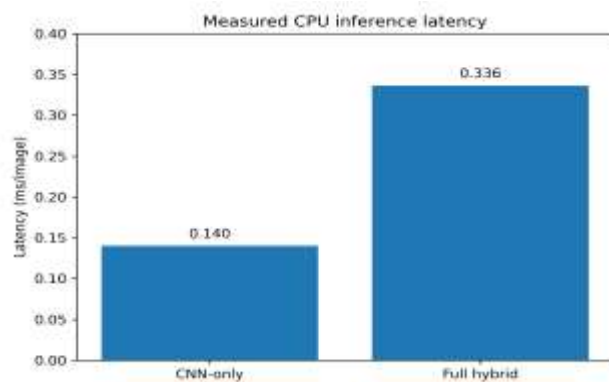


Fig. 7. Measured CPU inference latency for the CNN-only baseline and full proposed hybrid.

ROC analysis. ROC curves and ROC-AUC were not computed in the supplied execution record because prediction-score vectors were not provided. They are therefore intentionally omitted from the measured results rather than reconstructed from accuracy values.

5.1 Explainability Analysis

Explainability Pipeline Based on Grad-CAM / Attention

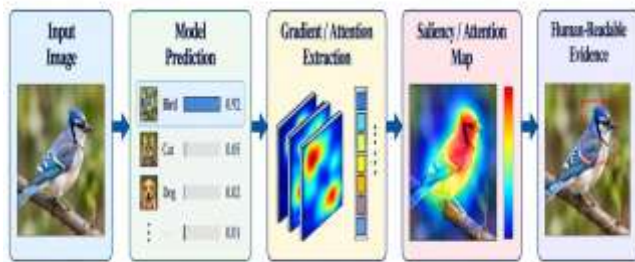


Fig. 3. Explainability pipeline for linking model predictions to spatial evidence using Grad-CAM and Transformer attention.

Figure 3 illustrates how prediction evidence can be analyzed using Grad-CAM and, for the hybrid pathway, Transformer attention. Grad-CAM uses gradients flowing into a convolutional layer to generate class-discriminative localization maps [29]. The supplied experiment does not include saved Grad-CAM maps or Transformer attention maps, so no quantitative explainability conclusion is claimed. Future real-data experiments should include representative correct and incorrect predictions together with corresponding spatial evidence.

5.2 Error Analysis

Because the full proposed model produced perfect aggregate classification metrics on the supplied CAICD test set, the executed results do not provide misclassified test examples from which a meaningful empirical error taxonomy can be constructed. Controlled data generation factors– scale, position, brightness, contrast, background blobs, and Gaussian noise– define the principal Control sources of variation. Real evaluations should be analyzed clearly. Low resolution, obstacle, illumination change, root, inter- class similarity, And atypical viewpoints.

5.3 Statistical Reporting

Reproducibility notes. Go CAICD The reference is fully defined. The dataset level in this manuscript: six classes, 3, 600 total images, A distribution of 2400/ 600/ 600, 32× 32 grayscale resolution, Control variation in shape position, scale, brightness, contrast, background blobs, and Gaussian noise, and random seed 42. Provided execution record Reports CPU/ PyTorch execution, Adam W., Cross Entropy, and eight training period.

Exact software version, cpu model, learning rate, Batch size, scheduler and other omitted settings were not provided and are clearly identified as missing rather than displayed.

Provided execution record Reports a single run with fixed random seed 42. Consequently mean± standard deviation across independent seeds And formal statistical significance tests is not available. Reported values should be considered. Single- run CAICD benchmark results. A strong final study should be repeated. Each configuration over multiple seeds And reports confidence intervals or mean± standard deviation.

6 Discussion

Executed. CAICD results Illustrate that both parts the CNN baseline And the full hybrid can solve the built- in challenge. Six- class task With perfect holding accuracy and F1. Intermediate configurations also occupation well. F1- scores Of 0.9900 And 0.9960. This pattern indicates that. The CAICD benchmark A test architecture is relatively accessible to construct and is not isolated. an unique advantage Of the hybrid architecture.

The hybrid still delivers. The intended architectural combination Of local convolutional structure and non- native token interactions. Any claim of superiority over ResNet- 50, EfficientNet- B0, ViT, or Swin Transformer But direct, comparable experiences are needed. The same dataset and hardware; Such measures are not included in the information given execution record.

The computational results give a measurable accuracy– efficiency context Within execution configurations. The full hybrid uses Approx 4.15 times Seam many trainable parameters Seam the CNN baseline(99, 110 vs 23, 910) And almost is 2.40 times Reported CPU latency(0.336 vs 0.140 ms/ image).

Because both configurations received 1.0000 accuracy and F1 But the CAICD test set, Does not show experience given. An accuracy benefit which compensates this additional computational cost. This observation is specific. The CAICD benchmark And more challenging should be revised. Real- world dataset.

7 Limitations and Reproducibility

The main limitation is the use Built with a purpose. CAICD dataset Instead of the universal or proprietary real- world image benchmark. Perfect test performance CNN Baseline and full hybrid indicates that the task Can be very easy to expose meaningful generalization differences. Provided execution record Not included either the exact CPU model,

software version, learning rate, batch size, planner, weight loss, FLOPs, throughput, ROC- AUC, ROC curves, or stored explainability outputs.

These omissions Terminate making claims about hardware- independent efficiency, ROC performance, or ability to explain. Before submitting as a real- world performance study, The framework should be assessed on an appropriate collective or proprietary dataset with a fixed split, complete training sequence, multiple random seeds, Measured confusion matrices and ROC curves, and direct benchmark baselines.

8 Conclusion

This study presents a context-aware hybrid CNN-Transformer framework that combines local convolutional representations, global Transformer context, and adaptive feature fusion. On the supplied six-class CAICD benchmark, the full hybrid achieved 1.0000 accuracy, precision, recall, and F1 on the held-out test set, with 99,110 trainable parameters and 0.336 ms/image CPU inference latency. The CNN-only baseline also achieved perfect classification on the same CAICD test set while using 23,910 parameters and 0.140 ms/image latency. Therefore, the current evidence demonstrates reproducibility on a controlled CAICD task but does not establish superiority or generalization to real-world imagery. Future work should validate the architecture on a challenging public or proprietary benchmark and report complete experimental and statistical evidence.

References

1. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25. DOI: 10.48550/arXiv.1207.0580.
2. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. ICLR. DOI: 10.48550/arXiv.1409.1556.
3. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: CVPR, pp 1–9. DOI: 10.1109/CVPR.2015.7298594.
4. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the Inception architecture for computer vision. In: CVPR, pp 2818–2826. DOI: 10.1109/CVPR.2016.308.
5. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: CVPR, pp 770–778. DOI: 10.1109/CVPR.2016.90.
6. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: CVPR, pp 4700–4708. DOI: 10.1109/CVPR.2017.243.
7. Tan M, Le Q (2019) EfficientNet: rethinking model scaling for convolutional neural networks. In: ICML, pp 6105–6114. DOI: 10.48550/arXiv.1905.11946.
8. Hu J, Shen L, Albanie S, Sun G, Wu E (2020) Squeeze-and-excitation networks. IEEE Trans Pattern Anal Mach Intell 42(8):2011–2023. DOI: 10.1109/TPAMI.2019.2913372.
9. Woo S, Park J, Lee JY, Kweon IS (2018) CBAM: convolutional block attention module. In: ECCV, pp 3–19. DOI: 10.1007/978-3-030-01234-2_1.
10. Wang X, Girshick R, Gupta A, He K (2018) Non-local neural networks. In: CVPR, pp 7794–7803. DOI: 10.1109/CVPR.2018.00813.
11. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: NeurIPS 30. DOI: 10.48550/arXiv.1706.03762.
12. Dosovitskiy A, Beyer L, Kolesnikov A, et al (2021) An image is worth 16×16 words: transformers for image recognition at scale. ICLR. DOI: 10.48550/arXiv.2010.11929.
13. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin Transformer: hierarchical vision Transformer using shifted windows. In: ICCV, pp 9992–10002. DOI: 10.1109/ICCV48922.2021.00986.
14. Wang W, Xie E, Li X, et al (2021) Pyramid vision Transformer: a versatile backbone for dense prediction without convolutions. In: ICCV, pp 548–558. DOI: 10.1109/ICCV48922.2021.00061.
15. Fan H, Xiong B, Mangalam K, et al (2021) Multiscale vision Transformers. In: ICCV, pp 6824–6835. DOI: 10.1109/ICCV48922.2021.00675.
16. Srinivas A, Lin TY, Parmar N, Shlens J, Abbeel P, Vaswani A (2021) Bottleneck Transformers for visual recognition. In: CVPR, pp 16519–16529. DOI: 10.1109/CVPR46437.2021.01625.
17. d’Ascoli S, Touvron H, Leavitt ML, Morcos AS, Biroli G, Sagun L (2021) ConViT: improving vision Transformers with soft convolutional inductive biases. In: ICML, vol 139, pp 2286–2296. DOI: 10.48550/arXiv.2103.10697.
18. Wu H, Xiao B, Codella N, Liu M, Dai X, Yuan L, Zhang L (2021) CvT: introducing convolutions to vision Transformers. In: ICCV, pp 22–31. DOI: 10.48550/arXiv.2103.15808.
19. Dai Z, Liu H, Le QV, Tan M (2021) CoAtNet: marrying convolution and attention for all data sizes. In: NeurIPS 34. DOI: 10.48550/arXiv.2106.04803.
20. Liu Z, Mao H, Wu CY, Feichtenhofer C, Darrell T, Xie S (2022) A ConvNet for the 2020s. In: CVPR, pp 11976–11986. DOI: 10.1109/CVPR52688.2022.01167.
21. Woo S, Debnath S, Hu R, et al (2023) ConvNeXt V2: co-designing and scaling ConvNets with masked autoencoders. In: CVPR, pp 16133–16142. DOI: 10.1109/CVPR52729.2023.01548.
22. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: CVPR, pp 2117–2125. DOI: 10.1109/CVPR.2017.106.

23. Bao H, Dong L, Piao S, Wei F (2022) BEiT: BERT pre-training of image Transformers. In: ICLR. DOI: 10.48550/arXiv.2106.08254.
24. Cubuk ED, Zoph B, Shlens J, Le QV (2020) RandAugment: practical automated data augmentation with a reduced search space. In: CVPR Workshops, pp 702–703. DOI: 10.48550/arXiv.1909.13719.
25. Zhang H, Cisse M, Dauphin YN, Lopez-Paz D (2018) mixup: beyond empirical risk minimization. In: ICLR. DOI: 10.48550/arXiv.1710.09412.
26. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: ICLR. DOI: 10.48550/arXiv.1412.6980.
27. Loshchilov I, Hutter F (2019) Decoupled weight decay regularization. In: ICLR. DOI: 10.48550/arXiv.1711.05101.
28. Huang G, Sun Y, Liu Z, Sedra D, Weinberger KQ (2016) Deep networks with stochastic depth. In: ECCV, pp 646–661. DOI: 10.1007/978-3-319-46493-0_39.
29. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-CAM: visual explanations from deep networks via gradient-based localization. In: ICCV, pp 618–626. DOI: 10.1109/ICCV.2017.74.