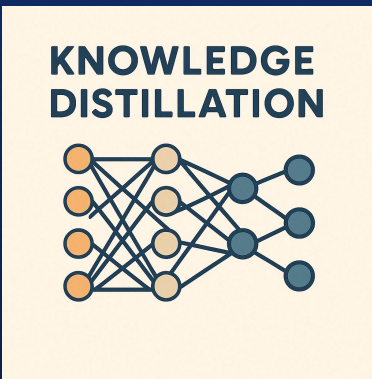


Knowledge Distillation

Presented by:
Yuval Luria
Roei Aviv
Omer Ben Simon





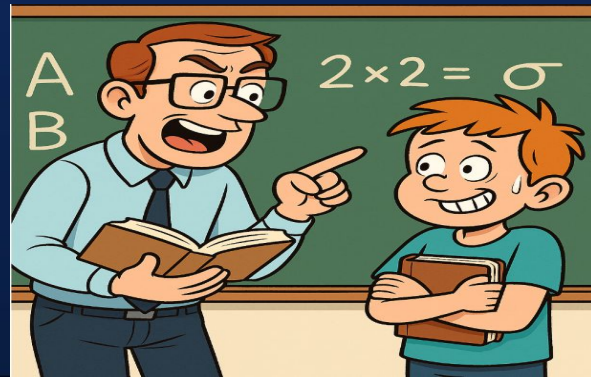
Abstract



Knowledge Distillation – Game Changer

KD is a model compression technique that enables the transfer of learned knowledge from a large, powerful model (called the Teacher) into a smaller, lightweight model (called the Student).

While the Teacher is trained to achieve high accuracy and complex decision boundaries, the Student learns to approximate the Teacher's behavior using a combination of hard labels (true class labels) and soft targets (probabilistic outputs of the Teacher).



From Giant to Mini: Teacher vs Student

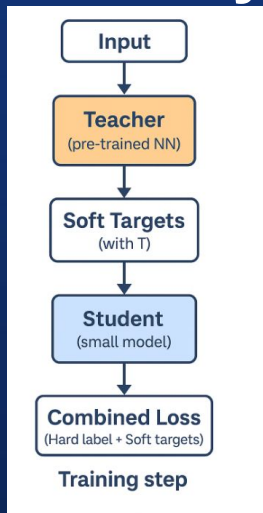
TEACHER vs STUDENT

TEACHER	STUDENT	
Model size	Hundreds of millions to billions of parameters	Few millions to hundreds of millions
Layers (Depth)	Many layers (e.g. 24-96)	Fewer layers (e.g. 4-12)
Hidden size (Width)	Large hidden dimensions	Smaller hidden dimensions
Attention heads (in Transformers)	Many heads (e.g. 16-64)	Fewer heads (e.g. 4-8)
FLOPs (Computations)	Very high	Much lower
Memory requirement	Huge (GPUs, TPUs)	Fits mobile/ edge devices
Inference speed	Slow (without strong hardware)	Fast (suitable for real-time)
Deployment target	Data centers (cloud)	Edge devices (phones, IoT, browsers)

The Student model is a compact version, optimized to capture the essential knowledge from the Teacher while being much smaller, faster, and more efficient for real-world deployment.

The Student doesn't try to copy everything the Teacher knows - it focuses on what's truly useful for deployment.

The way Knowledge Distillation Works:



1. Input is sent to a large, trained model (Teacher).
2. The Teacher outputs logits, then uses softmax with temperature to produce soft targets (class probabilities).
3. The same input goes to a smaller model (Student) which produces its own predictions.
4. A combined loss is calculated:
 1. One part compares with the true label.
 2. The other compares with the Teacher's soft targets.
5. The Student learns to mimic the Teacher while being smaller and faster.

•What is T (temperature)?

A scaling factor that controls how "soft" the probability distribution is.

•What does it do?

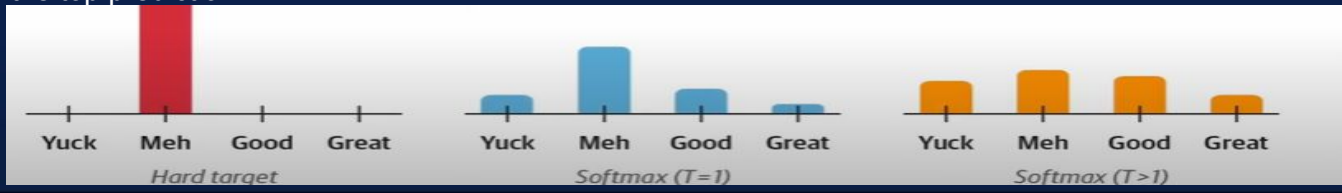
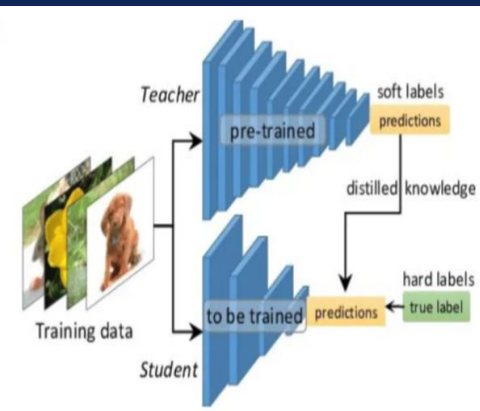
• $T=1$: standard softmax

• $T>1$: softer distribution (flatter, more information about non-top classes)

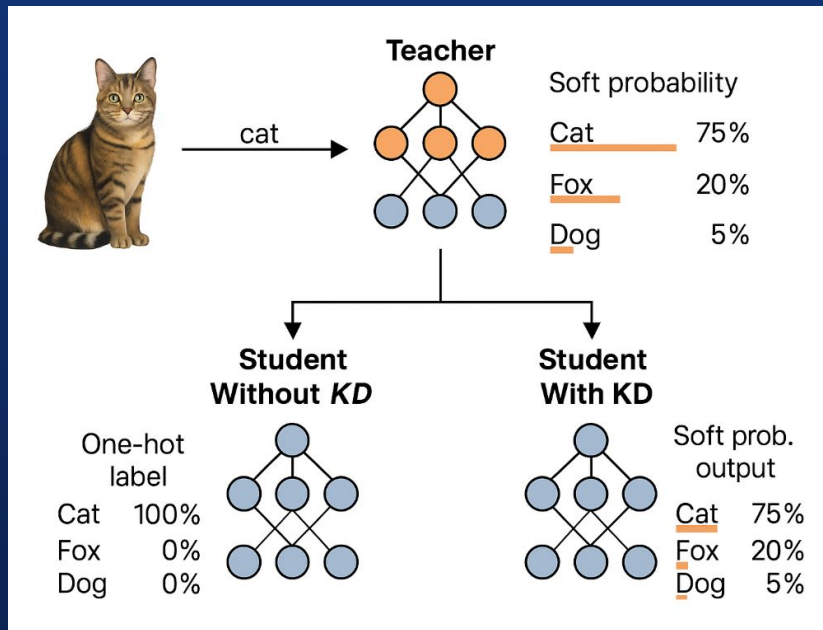
• $T<1$: sharper distribution (closer to one-hot)

•Why is it used in KD?

It helps expose the relative confidence of the Teacher across all classes — not just the top prediction.



What Happens Without KD vs. With KD



Student Without KD:

A student model trained without KD only learns from the final, correct label — as if it's the only truth.

For example, if the correct label is “cat,” the student is told:

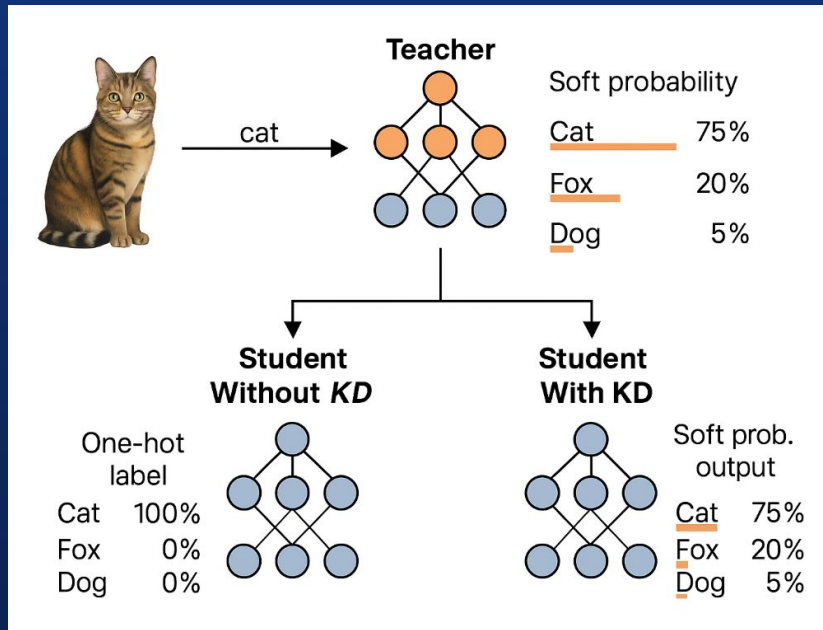
“It’s a cat. That’s it. Everything else is wrong.”

So the model only learns to map input images to one-hot labels, like:

- Cat = 100%, Dog = 0%, Fox = 0%

It has no idea that fox and cat might actually look somewhat similar.

What Happens Without KD vs. With KD



Student With KD:

Now imagine the same image goes through a Teacher model. Instead of just saying “cat,” the teacher gives a soft probability distribution:

- Cat = 75%
- Fox = 20%
- Dog = 5%

This helps the student understand that the model sees fox as a close second — and dog is far less likely.

So the student learns:

“If something looks 75% like a cat but 20% like a fox, I should be cautious. Maybe they share features.”

This is the magic of soft targets — the student doesn’t just memorize the answer, it learns how the teacher thinks.

Why It Matters:

Without KD: Student only learns “right or wrong”

With KD: Student learns relationships between classes

This helps with generalization — the model performs better on difficult or ambiguous inputs

From Logits to Probabilities – Softmax Explained

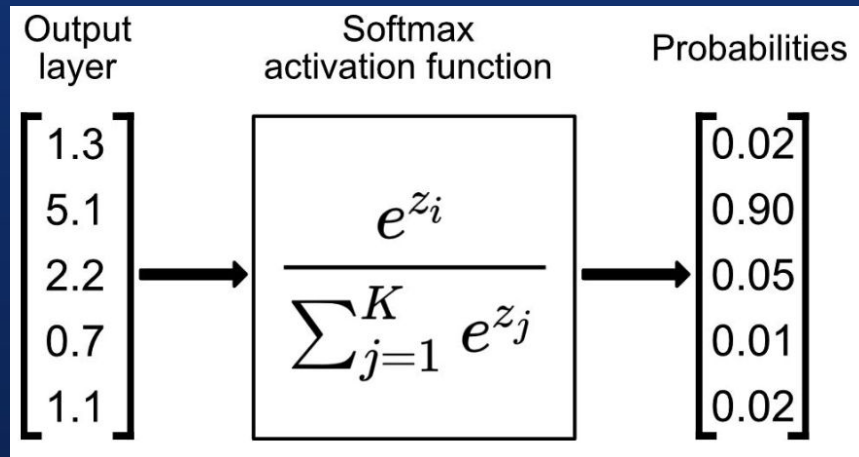
Step 1: Compute exponentials of each logit

Step 2: Compute the sum of exponentials

Step 3: Calculate each probability

Final probabilities (normalized)

Softmax transforms logits into probabilities by exponentiating each logit, normalizing them by the sum of all exponentials, so that the final outputs are positive and sum up to 1.



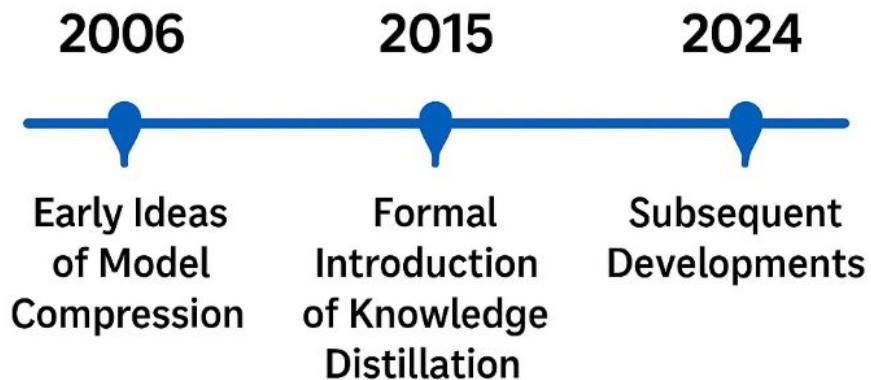


History



History

Knowledge Distillation: A Historical Perspective



2006 – Early Ideas

Geoffrey Hinton introduces the concept of “Dark Knowledge” — realizing that softmax outputs contain useful information beyond the correct label.

2015 – Formal KD Definition

Hinton, Vinyals & Dean formalize KD: a large Teacher model transfers knowledge to a smaller Student using soft and hard labels.

2024 – Advanced Developments

Emerging methods like Feature Distillation, Self-Distillation, and Online KD. Used in real-world models like DistilBERT and Tiny-YOLO.



Introduction



Introduction To Knowledge Distillation



Which Learning Type is KD:

Supervised Learning

- The Teacher is trained on labeled data.
- The Student learns from both:
 - The true labels (hard labels)
 - The Teacher's predictions (soft labels).

In most KD setups, supervision is still required — because the student learns both from the teacher's knowledge and from the real ground truth. The teacher guides the student's thinking, but the ground truth helps to keep the student grounded and avoid copying the teacher's mistakes.

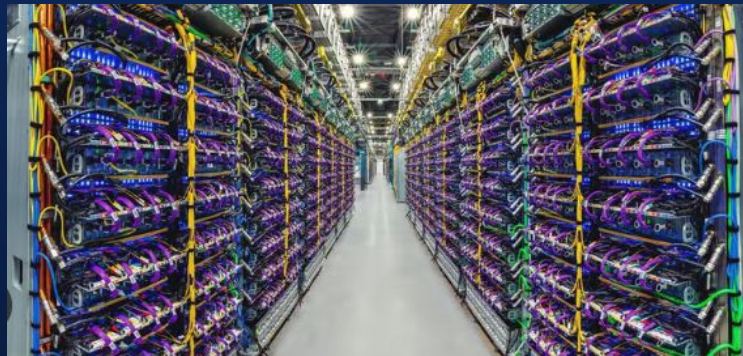


Why Do We Need Knowledge Distillation

As artificial intelligence advances, modern deep learning models continue to grow in size and complexity.

Models like GPT, BERT, and ResNet have achieved remarkable results — but with a cost..

- they are computationally expensive.
- consume vast memory.
- often unsuitable for deployment on resource-constrained environments such as mobile phones .





Object Detection – Before and After Knowledge Distillation

Object Detection Before KD:

The Problem:

large and heavy models like:

- Faster R-CNN
- YOLOv3
- RetinaNet

Drawbacks:

- High GPU and memory requirements
- Not suitable for real-time or mobile applications
- Couldn't run on phones, drones, or smart cameras

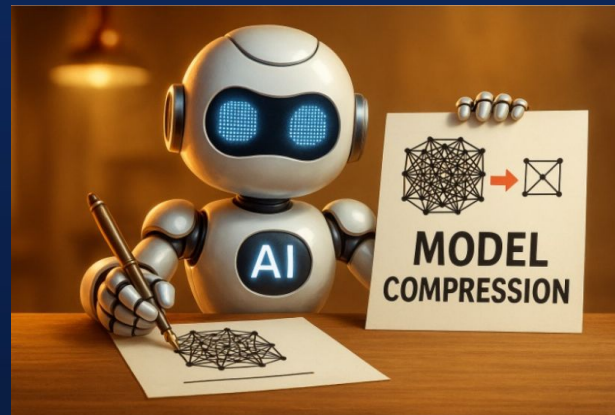
The Attempted Solution:

Lighter Models were developed

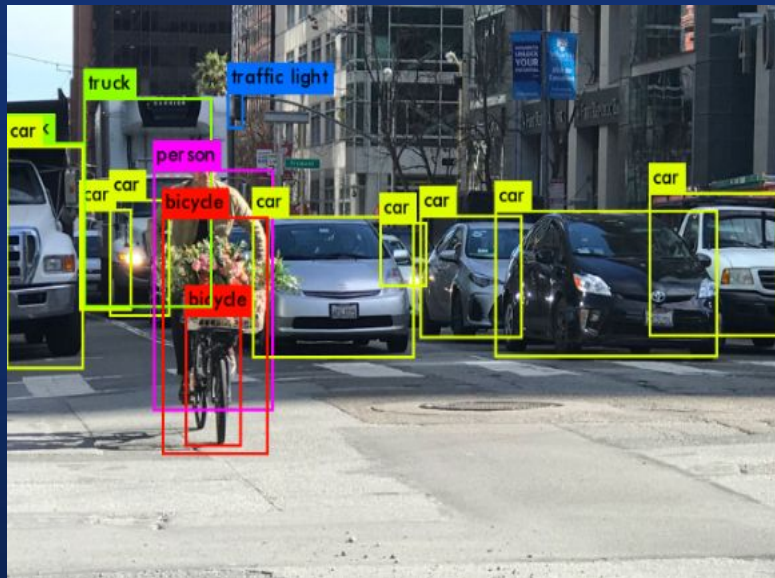
- Tiny-YOLO, MobileNet-SSD

But these suffered from:

- Lower accuracy
- Difficulty detecting small, occluded, or overlapping objects
- Poor performance in crowded or complex scenes



How Hard Was This Before Modern Models & KD?



Multi-Step, Complex Pipelines

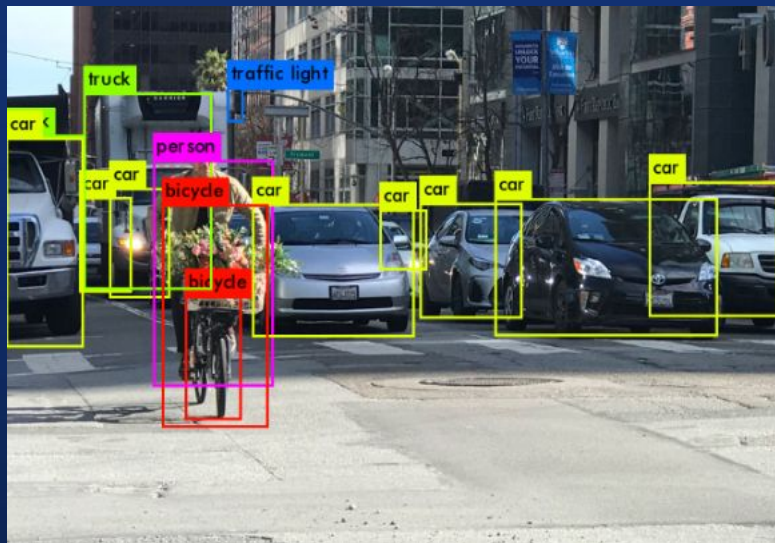
Before modern object detection models like YOLO and DETR:

- Detection used multi-stage pipelines like Faster R-CNN.
- These pipelines required:
 - Region Proposal Networks (RPNs).
 - Feature extraction and classification.
 - Manual post-processing (e.g., Non-Maximum Suppression).

This made them:

- Heavy (many parameters).
- Slow to train.
- Data-hungry.
- Difficult to deploy on real-time or edge devices.

How Knowledge Distillation Helped:



KD allows large models (teachers) to transfer “soft” knowledge to smaller models (students).

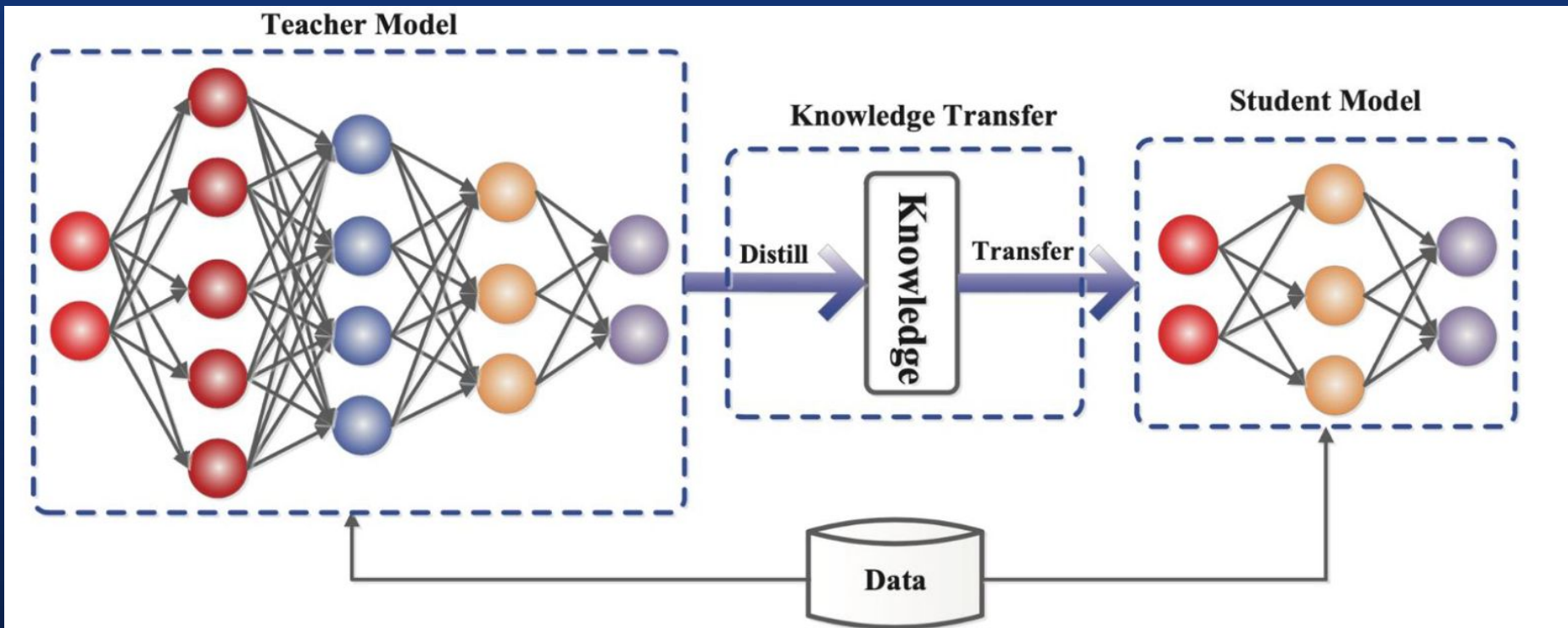
- Students learn not only the correct label (e.g., "bicycle"), but also relational cues:
 - Bicycle is near a person.
 - It's partially occluded.
 - It's on the road and moving.

This enables:

- Better generalization.
- Faster convergence.
- Compact models that work well even with fewer training examples.

- Detecting many objects with different sizes and overlaps was very hard before.
- Without KD or Transformer-based models, detection required heavy computing, large datasets, and complex pipelines.
- Today, KD allows small, efficient models to achieve high-quality detection like this — faster, cheaper, and smarter.

Architecture of Knowledge Distillation



Architecture of Knowledge Distillation – Teacher

- A large, pre-trained and high-capacity model (e.g., ResNet-152, BERT, DETR).
- Produces soft labels (probability distributions) for input data.
- Optionally outputs intermediate features (e.g., hidden states, attention maps).

Architecture of Knowledge Distillation – Distillation Loss Functions

- Combines different loss terms
 - Soft target loss:
 - KL divergence between student and teacher soft outputs.
 - Often includes temperature scaling to smooth distributions.
 - Hard label loss:
 - Cross-entropy loss between student output and ground-truth.
 - Feature-based loss (optional):
 - L2 loss or attention loss between intermediate teacher and student features.

1st paper – Focal and Global Knowledge Distillation for Detectors

Background & Motivation

- Knowledge Distillation (KD) has been effective in image classification but faces challenges in object detection due to:
 - Foreground-background imbalance in object detection tasks.
 - Feature discrepancies between teacher and student models, especially in different image regions.
- Equal treatment of all regions during distillation can lead to suboptimal performance.

1st paper – Focal and Global Knowledge Distillation for Detectors

Key Contributions

1. Focal Distillation

1. Separates foreground and background regions.
2. Encourages the student model to focus on the teacher's critical pixels and channels.
3. Utilizes attention masks to emphasize important areas, reducing the negative impact of feature disparities.

2. Global Distillation

1. Reconstructs the relationships between different pixels.
2. Transfers comprehensive spatial information from teacher to student.
3. Compensates for missing global context in focal distillation.


3. Unified Focal and Global Distillation (FGD) Framework

1. Combines focal and global distillation methods.
2. Applies loss calculations on feature maps, making it adaptable to various detectors.



1st paper – Focal and Global Knowledge Distillation for Detectors (Presented at CVPR 2022 by Zhendong Yang et al.)

Experimental Results

- Evaluated on multiple DETR variants, including RT-DETR with ResNet18.
 - Tested on two public datasets, including VisDrone.
 - Achieved:
 - +3.6% mAP@50 improvement for the student model.
 - +2.5% mAP@50–90 improvement.
 - In some cases, the student even outperformed the teacher (e.g., in mAP@50).
- 

2nd paper – Learning Efficient Object Detection Models with KD (NeurIPS 2017)



Background & Challenges

- Modern CNN-based object detection models are highly accurate but computationally intensive.
- This makes them unsuitable for real-time applications on limited-resource devices.
- Existing model compression often leads to significant drops in accuracy, especially for complex tasks like object detection.



2nd paper – Learning Efficient Object Detection Models with KD (NeurIPS 2017)

Main Contributions

1. End-to-End Distillation Framework

1. Introduces an integrated learning framework combining Knowledge Distillation (KD) and Hint Learning.
2. A compact student model is trained to mimic a larger, more accurate teacher model.

2. Customized Loss Functions

1. Weighted Cross-Entropy Loss:
 1. Addresses class imbalance, especially for the dominant background class in detection tasks.
2. Restricted Regression Loss:
 1. Guides the student in learning precise bounding box regression from the teacher.

3. Feature Hint Adaptation Layers

1. Adds adapter layers to help the student model learn from the intermediate features of the teacher.
2. Aligns feature spaces between teacher and student for effective hint transfer.

3rd paper - Distilling the Knowledge in a Neural Network Hinton et al. (2015)

Core Idea of Knowledge Distillation (KD)

- Proposes a framework where a small student model learns from a large teacher model.
- Teacher outputs soft probabilities (via softmax with temperature) that guide the student.
- The student is trained using a combined loss: soft teacher loss + hard label loss.

3rd paper - Distilling the Knowledge in a Neural Network Hinton et al. (2015)

Technical Insights

- Uses temperature scaling ($T > 1$) to soften predictions.
- Student mimics the teacher's behavior rather than just matching labels.
- Works across many tasks: speech recognition, vision, NLP.

3rd paper - Distilling the Knowledge in a Neural Network Hinton et al. (2015)

Practical Advantages

- Student models are significantly smaller and faster.
- KD provides better generalization and robustness than training with labels alone.
- Foundation of many modern model compression techniques.

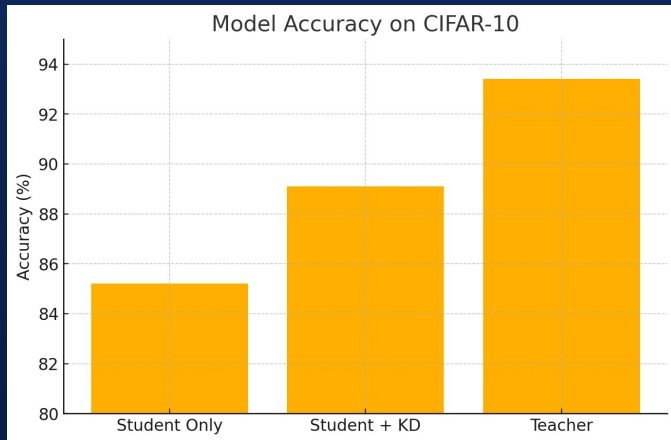
3rd paper - Distilling the Knowledge in a Neural Network Hinton et al. (2015)

Compare performance between Teacher, Student, and Distilled Student

Knowledge Distillation improves student accuracy by +3.9% without increasing size or latency

Student retains efficiency while approaching teacher-level accuracy

Ideal for deployment on edge devices



Model	Accuracy (%)	Model Size (MB)	Inference Time (ms)
Student Only	85.2	11	12
Student + KD	89.1	11	12
Teacher Model	93.4	25	30



4th paper - Hierarchical multi-attention transfer for knowledge distillation Guo et al., ACM (2022)


Problem:

- Vanilla Knowledge Distillation (KD) focuses on aligning logits or shallow features.
- This neglects the hierarchical and multi-level attention nature of deep networks.
- Attention-based KD improves performance, but is often shallow or layer-limited.

Goal:

Enhance student learning by transferring multi-level attention maps.

Capture both global-to-local and shallow-to-deep knowledge from the teacher





4th paper - Hierarchical multi-attention transfer for knowledge distillation Guo et al., ACM (2022)

Key Components:

Attention Extraction: From multiple layers using spatial-sum of squared features.

Hierarchical Grouping: Layers are organized into *stages* (e.g., conv1, conv2, conv3).

Multi-Attention Loss: MSE between normalized student and teacher attention maps per stage.





4th paper - Hierarchical multi-attention transfer for knowledge distillation Guo et al., ACM (2022)

Core Idea:

Introduce a Hierarchical Multi-Attention Transfer mechanism that:

- Extracts multi-level attention maps.
- Aligns them using a Hierarchical Attention Loss.
- Supports both cross-layer and cross-scale knowledge transfer.





4th paper - Hierarchical multi-attention transfer for knowledge distillation Guo et al., ACM (2022)

Results obtained :

Table 1. The Top-1 Accuracy (%) of HMAT (Ours) and Other Comparison Methods on CIFAR-10 Dataset

Teacher	ResNet34	ResNet101	ResNet18	WRN-28-10	ResNet18	ShuffleNetV2
Student	ResNet18	ResNet50	ShuffleNetV2	WRN-16-2	ResNet18	ShuffleNetV2
Baseline_T	95.39	95.55	95.01	95.98	94.93	91.03
Baseline_S	94.93	95.10	91.03	93.54	94.93	91.03
KD	95.03 (↑ 0.10)	95.19 (↑ 0.09)	91.80 (↑ 0.77)	93.63 (↑ 0.09)	94.96 (↑ 0.03)	92.19 (↑ 1.16)
AT	95.10 (↑ 0.17)	95.22 (↑ 0.12)	93.12 (↑ 2.09)	93.91 (↑ 0.37)	95.00 (↑ 0.07)	91.19 (↑ 0.16)
H-AT	95.09 (↑ 0.16)	95.12 (↑ 0.02)	91.77 (↑ 0.74)	93.57 (↑ 0.03)	94.97 (↑ 0.04)	91.04 (↑ 0.01)
CCKD	95.08 (↑ 0.15)	95.25 (↑ 0.15)	92.45 (↑ 1.42)	93.59 (↑ 0.05)	95.16 (↑ 0.23)	92.48 (↑ 1.45)
VID	95.19 (↑ 0.26)	95.28 (↑ 0.18)	92.03 (↑ 1.00)	93.70 (↑ 0.16)	95.17 (↑ 0.24)	92.41 (↑ 1.38)
KDAFM	95.07 (↑ 0.14)	95.26 (↑ 0.16)	–	93.84 (↑ 0.30)	95.01 (↑ 0.08)	–
CRD	95.06 (↑ 0.13)	95.13 (↑ 0.03)	91.95 (↑ 0.92)	–	94.99 (↑ 0.06)	91.51 (↑ 0.48)
HMAT	95.53 (↑ 0.60)	95.42 (↑ 0.32)	93.58 (↑ 2.55)	94.36 (↑ 0.82)	95.24 (↑ 0.31)	92.61 (↑ 1.58)


The symbol ↑ means the improvement over the baseline of the student network.

Acronym	Full Name
KD	Knowledge Distillation (vanilla, from Hinton et al.)
AT	Attention Transfer
H-AT	Hierarchical Attention Transfer (not HMAT, simpler version)
CCKD	Correlation Congruence for KD
VID	Variational Information Distillation
KDAFM	KD via Adversarial Feature Manipulation
CRD	Contrastive Representation Distillation



Research Question:

Can the incorporation of Alignment and Attention Transfer techniques enhance the effectiveness of Knowledge Distillation in improving student model accuracy for image classification?

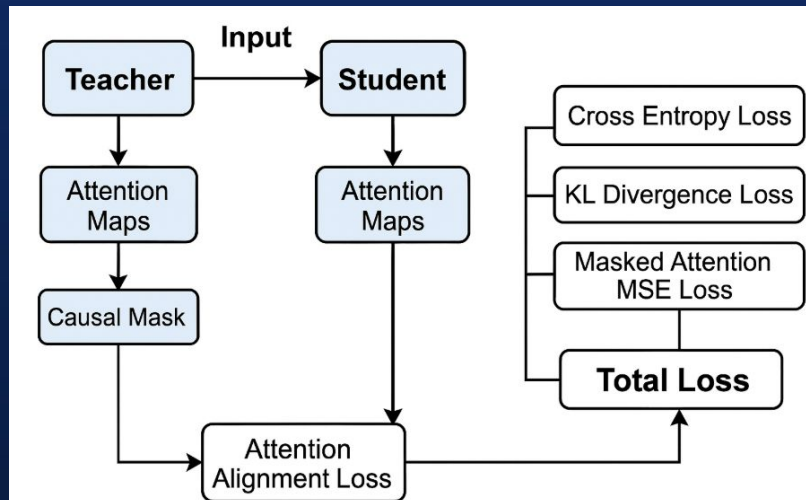


What Makes Our Method Distinct

Instead of transferring all attention features, we selectively transfer only the causally influential attention maps from a teacher to a student

Our goals are to:

- Disentangle the effects of Alignment and Attention Transfer in Knowledge Distillation
- Greater model efficiency





Attention maps

Attention Maps in KD – Key Ideas

- What is attention? A spatial map indicating which parts of the input the network focuses on.
- How it's used: Extracted by computing the sum of squared feature activations across channels.
- Why it matters: Transferring attention maps helps the student learn where to focus, mimicking the teacher's internal reasoning.



Alignment in KD

- **What is alignment?** A contrastive learning mechanism that encourages the student's attention to match the teacher's only for the same image.
- **How it's used:** Apply contrastive loss: pull positive pairs (same image) together, push negative pairs (different images) apart in the attention space.
- **Why it matters:** It avoids blind copying — the student learns to focus like the teacher, but only when appropriate.

What Are We Changing and Exploring?

- Changing:
 - Replacing standard training using ensemble methods with Knowledge Distillation
 - Student model learns from both labels and the soft predictions of the Teacher
- Research Focus:
 - How much accuracy can be gained by using KD?
 - Can a smaller student reach close to the teacher's performance?
 - How do intermediate features help in Hierarchical KD?

DataSet





CIFAR-10 – Our Dataset

Contains 60,000 color images (32x32 pixels)

50,000 for Training, 10,000 for Validation

10 categories: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck

Challenges:

Small image resolution

Low intra-class variance, high inter-class similarity





Knowledge Distillation on CIFAR-10

Goal: Improve the performance of a lightweight neural network using knowledge distilled from a larger, well-trained model.

Method: Knowledge Distillation (KD) – transferring knowledge from a large Teacher model to a smaller Student model.

Dataset: CIFAR-10 – a well-known image classification benchmark.

Focus: Accuracy, efficiency, and generalization.






Methodology

1. Dataset and Preprocessing

- **Dataset:** CIFAR-10 – 60,000 color images (32x32), 10 balanced classes.
- **Split:** 50,000 training / 10,000 test images.
- **Preprocessing:**
 - Random horizontal flip
 - Random crop with padding
 - Normalization using dataset mean and std

2. Models Setup

- **Teacher:** Pretrained ResNet50 with CBAM (Convolutional Block Attention Module) to enhance spatial+channel attention.
 - **Student:** Lightweight ResNet18 trained from scratch.
- 

Methodology

3. Causal Attention Map Extraction

- Use Integrated Gradients (IG) on the teacher model to compute per-sample causal importance over intermediate feature maps (e.g., layer3).
- Apply IG masks on the attention maps to obtain only the causally-relevant regions for each sample.

4. Distillation Loss Components

- **Cross Entropy Loss (CE):**
Supervision from true labels — helps the student learn to predict the correct class.
- **KL Divergence Loss (KL):**
Soft target distillation — aligns student's class probabilities with the teacher's.

Methodology

5. Training Strategy

- Forward input batch through teacher → extract attention + causal mask
- Forward same batch through student
- Compute combined loss:
TotalLoss =
$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{KD}} + \beta \cdot \mathcal{L}_{\text{Align}} + \gamma \cdot \mathcal{L}_{\text{Attn}}$$
- Backpropagate and update student

6. Evaluation Metrics

- Top-1 Accuracy on CIFAR-10 test set
- Model Size (params)
- Inference Time

Questions ?

