Automatically Extracting Scientific Metrics with LLMs: A Case Study of ImageNet Papers

Anonymous Author(s)

ABSTRACT

We collect a large dataset of publications that report Top-1 Accuracy on the ImageNet dataset and manually annotate it.

We report on automatically extracting the metrics using an LLM, and present a qualitative error analysis on a large development set.

ACM Reference Format:
Anonymous Author(s). 2018. Automatically Extracting Scientific Metrics with LLMs: A Case Study of ImageNet Papers. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

State-of-the-art (SOTA) performance metrics reported in research publications play a central role in benchmarking model progress, comparing results, and conducting meta-analyses [1, 2, 6]. However, the task of extracting performance metrics from scientific papers remains largely manual, error-prone, and inconsistent. This is due to the wide variation in reporting styles, ambiguous metric descriptions, and the frequent use of error rates or qualitative statements in place of clearly labeled accuracy values.

In this work, we present *EXTRACT-AND-VERIFY*, an end-to-end pipeline that leverages large language models (LLMs) to automatically extract scientific metrics from publications, with a focus on those that evaluate on the ImageNet dataset. We construct and publicly release a development set comprising 100 ImageNet-related papers, each manually annotated with verified Top-1 accuracy values.

Our pipeline consists of a prompt-based *EXTRACT-AND-VERIFY* loop that filters and rechecks extracted metrics across multiple passes. When tested on the proposed annotated development set, *EXTRACT-AND-VERIFY* successfully retrieves the correct Top-1 accuracy in 67 of the 100 papers, outperforming the SCILEAD baseline, which succeeds in 63 cases. By releasing both our annotated dataset and the full pipeline, we provide a reproducible benchmark for future work on automatic accuracy extraction and LLM-assisted literature analysis.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

2 RELATED WORK

2.1 Prompting

Few shot/In-Context Learning (ICT) [3]. Few-shot prompting (also referred to as "in-context learning") provides a model with a prompt consisting of k in-context examples of the target task, each in the form of input-output pairs ⟨xi , yi ⟩ . Few-shot (FS) is a prompting strategy where models are given demonstrations of the task at inference time as conditioning but no weights are updated from. Each demonstration of the the task consists of a context and a desired completion. The work argues that FS brings the benefits for reducing task-specific data however may also producing worser performances than existing SOTA models. The categories of the FS schemas are as followed:

- Zero-shot (0S) similar to FS but with a natural language description of the task instead of any examples
- One-Shot (1S) - similar to FS but with K = 1
- Context window needs to be investigated

Iterative Refinement with SELF-REFINEMENT [5]. The approach, SELF-REFINEMENT, involves query self-refinement and claims to enhance the initial outputs of large language models (LLMs). Self-refinement is inspired by the way how humans revise their written text. This method consist of an iteratively feedback-driven process that improves the initial responses generated by LLMs.

In the following, we will begin by examination the examples of the p_gen, p_fb, prefine prompts for relevant tasks. Their key idea is that SELF-REFINE uses the same underlying LLM to generate, get feedback, and refine its outputs given its own feedback. It relies only on supervision present in the few-shot examples.

The work evaluates empirically on specific task datasets, such as dialogue response generation, demonstrating its effectiveness in refining initial results from LLMs. In our context, this work suggests a method for refining queries to LLMs, enhancing the quality and relevance of their outputs. Notably, the metric associated with Dialogue Response Generation involves Human-pref. That is a blind human A/B evaluation on a subset of the outputs to select the preferred output. Beyond human-pref, the work use GPT-4 for human preference resulting in 71% on the task. [Need to check the references] Their results on Dialogue Response Generation leads to high gains, and improvements by 49.2% – from 25.4% to 74.6% on GPT-4 preference scores.

Different from the QA/RAG point of view whose focus is on constructing the relevant knowledge base for LLM, in order to improve the performance, SELF-REFINEMENT provides its insights from feedback to LLM.

2.2 Emphirical Methods: SCILEAD [7]

Recent work, SCILEAD on scientific leaderboards produces ranking systems and extracting task, dataset and evaluation metric (TDM) triples. The work utilizes a recent large-language model and addresses these limitations by offering a manually curated dataset of leaderboards drawn from 43 scientific papers. The work exhaustively annotates each paper for all unique TDM combinations, along with their top-reported results. Interestingly, SCILEAD's curated approach and flexible framework presents an automated construction of scientific leaderboards and benchmarks.

2.3 Self-Critique

Several recent studies have explored the concept of direct self-critique by large language models (LLMs) [4, 8, 9]. Inspired by human cognitive processes, these approaches leverage the intuition that verifying or critiquing an answer is typically easier or fundamentally different from generating it initially, thereby potentially improving the overall output quality [8]. In a standard self-critique pipeline, an LLM first generates an answer and subsequently receives its own response as input, along with explicit instructions prompting it to critique, refine, or revise the original answer. This self-critique loop continues iteratively until reaching a predefined stopping criterion.

Our EXTRACT-AND-VERIFY pipeline builds upon this idea by incorporating explicit verification steps, enabling cross-checking and refinement of extracted state-of-the-art metrics.

3 ANNOTATED DEVELOPMENT SET

In the section, we describe the construction of our development set. We begin with the automated crawling pipeline that fetched candidate PDFs, then describe the manual annotation protocol and finally discuss how metric selection and dataset-subset alignment were harmonised across all papers.

3.1 Paper Collection

Our dataset originates from an automated collection of publication entries through the PaperWithCode platform, focusing on computer vision papers that report results on ImageNet dataset. We implemented a simple "try/catch" script to repeatedly query PaperWithCode for PDFs or arXiv links, automatically skipping entries that lacked valid URLs or produced parsing errors. This step yielded an initial pool of candidate papers, all of which claimed to present Top-1 accuracy on ImageNet or its widely recognized variants. We programmatically retrieved ImageNet image-classification papers using the `paper_dataset_list` endpoint of the PaperWithCode API, which returns results in its default (unspecified) order.[1] We first selected 12 papers to tune our prompts and then curated another 100 papers to build our development set.

3.2 Label-Verification Protocol

From the successfully retrieved documents, we curated a corpus of papers under the theme of ImageNet-based classification. To ensure correctness, we performed a manual verification of each paper's reported performance metrics. Specifically, we identified references to "Top-1 Accuracy," pinpointed the corresponding numerical values. We also recorded the precise page or section where these metrics appeared, thereby creating explicit labels: *(Dataset: ImageNet, Metric: Top-1 Accuracy, Location: Page)*. This step was crucial to catch discrepancies, such as papers mentioning alternative datasets or partial results that do not correspond directly to the final accuracy numbers.

			Test		10-crop Validation	Single-Model Validation	
File Name	Paper Name	Model	Top-1	Top-5 A	Top 1	Top-1	Top-5 A
1512.03385v1.pdf	Deep Residual Learning for Image R	ResNet-18					
1512.03385v1.pdf	Deep Residual Learning for Image R	ResNet-34					
1512.03385v1.pdf	Deep Residual Learning for Image R	ResNet-50					
1512.03385v1.pdf	Deep Residual Learning for Image R	ResNet-101					
1512.03385v1.pdf	Deep Residual Learning for Image R	ResNet-152		96.43%	78.57%	80.57%	
1703.09844v5.pdf	Multiscale Dense Networks for Resc	MSDNet	75%	-			-
1803.00942v3.pdf	Not All Samples Are Created Equal:	ResNet-50	-	-		-	-
1807.10108v5.pdf	Effects of Degradations on Deep Ne	V-CapsNet		-		99.83%	-
1807.10119v3.pdf	A Unified Approximation Framework	AlexNet	-	80%		-	
1807.11164v1.pdf	ShuffleNet V2: Practical Guidelines	ShuffleNet v2-50	77.20%	-			-
1807.11254v2.pdf	Extreme Network Compression via F	ResNet34	64.75	64.3			
1807.11459v1.pdf	Improving Transferability of Deep Ne	ResNet-27	-	-		-	-
1807.11626v3.pdf	MnasNet: Platform-Aware Neural Ar	MnasNet	75.20%	-			-
1909.11155v1.pdf	Anchor Loss: Modulating Loss Scale	ResNet-50	76.82%	93.03%			
1909.13863v1.pdf	XNOR-Net++: Improved Binary Neura	Binary ResNet-18	57.10%	79.90%			
1909.13863v1.pdf	XNOR-Net++: Improved Binary Neura	Binary AlexNet	46.90%	71.00%			
omniVec_2023.pdf	OmniVec: Learning robust represent	OmniVec (FT)	92.40%	-			-

Figure 1: Ground truth table for a selected set of 12+ papers, indicating whether Top-1/Top-5 results are from the test set, validation set, or a specific multi-crop procedure. "–" denotes that no explicit metric was identified for that field. Some entries include partial or approximate values extracted from the text, which may correspond to a specific subset or alternative challenge (e.g., 10-crop validation).

3.3 Dataset Alignment on ImageNet

Most papers in our corpus report results Many of the papers in our corpus rely on *variants* or *subsets* of the ImageNet dataset. For instance, it is common to see references to ILSVRC-2012 or ILSVRC-2015, each of which differ slightly in the number of classes or distribution of images. Moreover, a recurring trend is to report results on the *validation set* rather than the *held-out test set*, particularly in cases where official test-set submissions are limited or not feasible. Researchers sometimes further augment or preprocess validation data in ways that yield higher accuracies than standard evaluation protocols. As a result, metric comparisons across papers can be ambiguous without clarifying details on the exact subset of ImageNet or cropping strategies used.

In Fig. 1, we align each paper's reported metrics—often referencing *only* a validation subset or an aggressive multi-crop strategy—to a consistent schema. This approach helps unify the variety of reporting practices, though residual uncertainties (e.g., official test vs. internal hold-out sets) still pose challenges for direct comparison among the papers.

4 METHODOLOGY: EXTRACT-AND-VERIFY PIPELINE

In the section, we will discuss our methodology, EXTRACT-AND-VERIFY in detail. Our method employs a "*EXTRACT-AND-VERIFY*" approach.

[1] The API documentation does not state the sorting criterion; see https://paperswithcode-client.readthedocs.io/en/latest/api/client.html.

After prompting the LLM to extract performance metrics from different sections, the model is also required to cite the corresponding original sentence. The process is repeated for k iterations. During the verification stage, the LLM compares the results from the k iterations to verify the cited sentences and their corresponding accuracy metrics. It votes on the most commonly cited sentences and calculates the final accuracy based on these results. This self-verification process is applied across all chunks in the corpus. Chunks that do not contain accuracy figures are discarded. The relevant chunks are combined to deliver the final results. This process mimics the strategy humans often use when searching for relevant sections in academic papers, such as conclusions or experimental results. The rationale behind this approach is that academic papers typically follow a structured format, with key findings and state-of-the-art performance figures most likely reported in the conclusion or experimental sections.

EXTRACT-AND-VERIFY. Our final method, "*EXTRACT-AND-VERIFY*," leverages an iterative loop: the model first extracts potential accuracy metrics and cites the source sentence, then re-examines its own outputs to confirm consistency or highlight omissions. This verification step forces the LLM to reflect on potential errors—especially when multiple snippet extractions conflict. We found *EXTRACT-AND-VERIFY* particularly effective for discarding spurious references (e.g., baseline or dev-set accuracies) and ensuring that the extracted top-1 accuracy indeed matches the final reported performance on ImageNet.

Our prompt: Given the following *EXTRACT-AND-VERIFY* prompt:

""" extract the top1 accuracy of ImageNet from the given text and return both the sentence containing the accuracy. Answer in a number, eg. 90.2

Example 1: Expected Output: Sentence: "a table showing the Top-1 and Top-5 classification accuracy using a binarized ResNet-18 on Imagenet for various ways of constructing the scaling factor. The method "Case 4: " achieved a Top-1 accuracy of 57.1." Accuracy: 57.1

Example 2: Expected Output: Sentence: "In our experiments, the model reported a Top-1 accuracy of 82.1Accuracy: 82.1

Example 3: Expected Output: Sentence: "The evaluation results showed a Top-1 accuracy of 78.3Accuracy: 78.3

Example 4: Expected Output: Sentence: "Our proposed model achieved a Top-1 accuracy of 74.2Accuracy: 74.2

Example 4: Expected Output: Sentence: "Our proposed model achieved a top-5 accuracy of 66.2Accuracy: -

Now extract the top1 accuracy of ImageNet from the following texts, page

Expected Output: """

5 EXPERIMENTAL RESULTS AND DISCUSSION

5.1 Experimental Results: *EXTRACT-AND-VERIFY* vs. SCILEAD

We evaluate the performance of our proposed extraction pipeline, *EXTRACT-AND-VERIFY*, against the existing SCILEAD baseline on our full annotated development set described in Section 3.

5.1.1 Development Set Results. We evaluate our proposed *EXTRACT-AND-VERIFY* system against SCILEAD on our 100-paper development set. Two types of evaluation are conducted: (i) classification accuracy—assessing whether the predicted Top-1 value exactly matches the annotated ground-truth, and (ii) regression-based metrics—quantifying how close the extracted values are numerically.

Classification Metrics. We first evaluate the classification accuracy of both systems. Table 1 presents standard classification metrics, including accuracy, precision, recall, and F1 score.

System	Accuracy	Precision	Recall	F1 Score
EXTRACT-AND-VERIFY	0.670	1.000	0.670	0.802
SCILEAD	0.630	1.000	0.630	0.773

Table 1: Classification metrics on the 100-paper development set.

Our *EXTRACT-AND-VERIFY* pipeline correctly extracts the Top-1 accuracy in 67 out of 100 papers, compared to 63 by SCILEAD. However, our method yields higher recall and F1 score, reflecting better overall coverage and extraction consistency. These classification-based findings align with our regression analysis discussed below.

Regression Metrics. To further quantify performance, we compute regression metrics over valid samples—papers where both ground-truth and predictions are numeric. Table 2 presents the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Pearson Correlation.

System	#Samples	MAE	RMSE	Correlation
EXTRACT-AND-VERIFY	23	0.642	2.112	0.971
SCILEAD	12	21.247	49.429	-0.108

Table 2: Regression metrics comparing *EXTRACT-AND-VERIFY* and SCILEAD.

Our system shows a dramatically lower MAE and RMSE compared to SCILEAD, indicating tighter alignment with the annotated ground-truth values. The high Pearson correlation ($r = 0.971$) also suggests that our method better preserves the relative ranking of performance metrics across papers, whereas SCILEAD shows weak and even negative correlation.

5.2 Qualitative Analysis

We illustrate our results with a qualitative analysis of easy and difficult cases.

Straightforward ("Easy") Examples.

- **Abstract, 1807.11626v3** (Fig. 2): "*On the ImageNet classification task, our **MnasNet** achieves 75.2% top-1 accuracy . . .*" *—Single numeric value in the abstract; metric name and dataset are explicit.*
- **Main text, 1703.09844v5** (Fig. 3): "*MSDNet achieves a top-1 accuracy of ∼75% . . .*" *—Similar pattern in the body; the tilde does not cause the confusion.*

Difficult ("Challenging") Examples.

- **Alternate dataset variant** (Fig. 4): "*V-CapNet reaches 99.83% validation accuracy on the **Natural Images** dataset.*" —*Dataset differs from canonical ImageNet; the numeric value must be ignored.*
- **Table-only reporting** (Fig. 5): Table lists only error rates (e.g. 19.38% Top-1 err.) without corresponding accuracies. —*Requires error-to-accuracy conversion and table parsing.*
- **Hidden in large metric tables** (Fig. 6): ImageNet Top-1 accuracy (92.4%) appears as a single cell among many datasets and metrics. —*Requires locate the correct row/column and ignore unrelated numbers.*
- **Table-only Top-1 value** (Fig. 7): ImageNet Top-1 accuracy (76.82%) is given only in a table, with no reference in the main text. —*Requires table reading; plain text search may be error-prone.*
- **Multiple candidate Top-1 values** (Fig. 8): A table in 1909.13863v1 lists four *Case* rows with different Top-1 accuracies (55.5–57.1%). —*Require an investigation on which value is the main result or flag ambiguity.*
- **Top-1 omitted, only Top-5 present** (Fig. 9): Sentence in 1807.10119v3 reports "top-5 accuracy drops slightly …" while never stating Top-1. —*Require returning "missing" rather than hallucinate a Top-1 value.*

Key take-aways. Easy cases share three traits: a single accuracy figure, explicit dataset naming, and standard wording. Failures occur when authors (i) report on validation instead of test set, (ii) report on ImageNet variants or sampled subsets, (iii) place numbers only in supplementary material, or (iv) use alternate metrics such as error rates. These observations guided our rule-based post-processing (for error-to-accuracy conversion) and the heuristics that flag ambiguous dataset references.

6 CONCLUSION

We introduced *EXTRACT-AND-VERIFY*, an end-to-end pipeline built on large language models (LLMs) for automatically extracting Top-1 accuracy metrics from scientific papers. To support this goal, we curated and publicly released a development set of 100 human-annotated computer vision papers focused on ImageNet classification. The dataset is larger than the one used in prior work [7]. To our knowledge, we are the first to apply a self-verification loop to scientific metric extraction.

Evaluated on the development set, *EXTRACT-AND-VERIFY* correctly retrieved the Top-1 accuracy in 67 out of 100 papers, outperforming the SCILEAD [7], which succeeded in 63 cases.

Increasing our dataset size and incorporating multiple human-curated metrics from diverse domains would enable fine-tunining of metric extraction systems further.

By releasing both our annotated dataset and extraction pipeline, we aim to establish a reproducible benchmark for future work on LLM-assisted metric extraction and support broader research into automating scientific understanding from machine learning literature.

REFERENCES

[1] Erin S Barry, Jerusalem Merkebu, and Lara Varpio. 2022. Understanding state-of-the-art literature reviews. *Journal of graduate medical education* 14, 6 (2022), 659–662.

[2] Lutz Bornmann, Robin Haunschild, and Ruediger Mutz. 2021. Growth rates of modern science: A latent piecewise growth curve approach to model publication numbers from established and new literature databases. arXiv:2012.07675 [cs.DL] https://arxiv.org/abs/2012.07675

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL] https://arxiv.org/abs/2005.14165

[4] Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2023. Teaching Large Language Models to Self-Debug. arXiv:2304.05128 [cs.CL] https://arxiv.org/abs/2304.05128

[5] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-Refine: Iterative Refinement with Self-Feedback. arXiv:2303.17651 [cs.CL] https://arxiv.org/abs/2303.17651

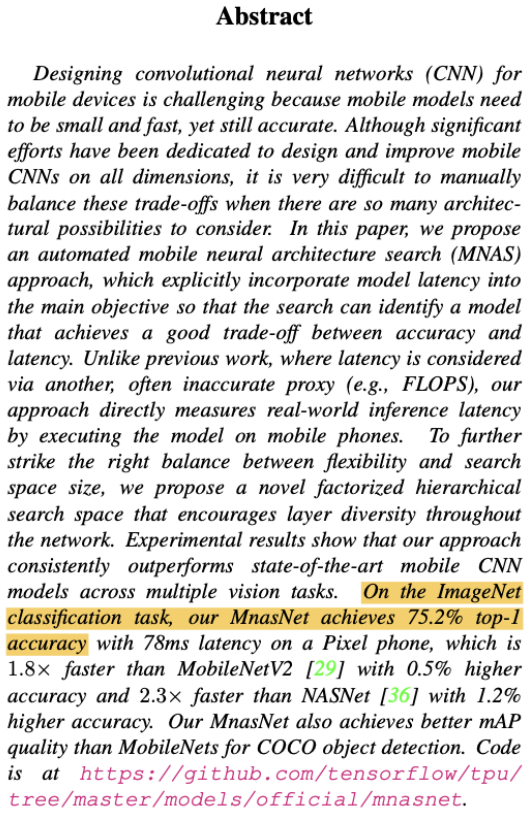
[6] Monica M. McGill, Tom McKlin, and Errol Kaylor. 2019. Defining What Empirically Works Best. *Proceedings of the 2019 ACM Conference on International Computing Education Research* (Jul 2019), 199–207. https://doi.org/10.1145/3291279.3339401

[7] Furkan Şahinuç, Thy Thy Tran, Yulia Grishina, Yufang Hou, Bei Chen, and Iryna Gurevych. 2024. Efficient Performance Tracking: Leveraging Large Language Models for Automated Construction of Scientific Leaderboards. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 7963–7977. https://doi.org/10.18653/v1/2024.emnlp-main.453

[8] Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. 2024. On the Self-Verification Limitations of Large Language Models on Reasoning and Planning Tasks. arXiv:2402.08115 [cs.AI] https://arxiv.org/abs/2402.08115

[9] Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large Language Models are Better Reasoners with Self-Verification. arXiv:2212.09561 [cs.AI] https://arxiv.org/abs/2212.09561

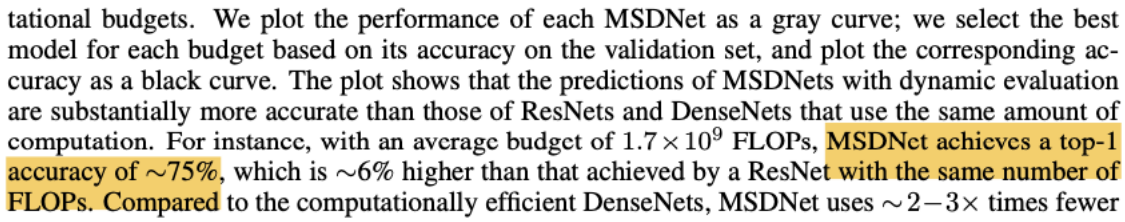
A SCREENSHOTS FROM ARXIX PAPERS WITH EXTRACT SECTIONS HIGHLIGHTED



Abstract

Designing convolutional neural networks (CNN) for mobile devices is challenging because mobile models need to be small and fast, yet still accurate. Although significant efforts have been dedicated to design and improve mobile CNNs on all dimensions, it is very difficult to manually balance these trade-offs when there are so many architectural possibilities to consider. In this paper, we propose an automated mobile neural architecture search (MNAS) approach, which explicitly incorporate model latency into the main objective so that the search can identify a model that achieves a good trade-off between accuracy and latency. Unlike previous work, where latency is considered via another, often inaccurate proxy (e.g., FLOPS), our approach directly measures real-world inference latency by executing the model on mobile phones. To further strike the right balance between flexibility and search space size, we propose a novel factorized hierarchical search space that encourages layer diversity throughout the network. Experimental results show that our approach consistently outperforms state-of-the-art mobile CNN models across multiple vision tasks. On the ImageNet classification task, our MnasNet achieves 75.2% top-1 accuracy with 78ms latency on a Pixel phone, which is 1.8× faster than MobileNetV2 [29] with 0.5% higher accuracy and 2.3× faster than NASNet [36] with 1.2% higher accuracy. Our MnasNet also achieves better mAP quality than MobileNets for COCO object detection. Code is at https://github.com/tensorflow/tpu/tree/master/models/official/mnasnet.

Figure 2: Easy example. Top-1 accuracy (75.2 %) is stated plainly in the abstract of *1807.11626v3.pdf*.



tational budgets. We plot the performance of each MSDNet as a gray curve; we select the best model for each budget based on its accuracy on the validation set, and plot the corresponding accuracy as a black curve. The plot shows that the predictions of MSDNets with dynamic evaluation are substantially more accurate than those of ResNets and DenseNets that use the same amount of computation. For instance, with an average budget of 1.7×10^9 FLOPs, MSDNet achieves a top-1 accuracy of $\sim$75%, which is $\sim$6% higher than that achieved by a ResNet with the same number of FLOPs. Compared to the computationally efficient DenseNets, MSDNet uses $\sim 2-3\times$ times fewer

Figure 3: Easy example. A single sentence in the main text of *1703.09844v5.pdf* reports Top-1 accuracy (75 %).

B. Analysis of network depth in CapsuleNet architecture

In our study, CapsuleNet shows significantly higher robustness against image degradation than conventional deep CNNs. However, state-of-the-art deep CNNs achieve better recognition accuracy than CapsuleNet for noise-free samples of all datasets. To improve the baseline performance of CapsuleNet, we introduce a novel fusion architecture ***V-CapsNet*** optimize the network by minimizing the marginal loss only. In our experiments, the proposed V-CapsNet fusion architecture achieves 99.83% validation accuracy on the *natural images* dataset, improving the baseline performance of CapsuleNet by 6.2%. Fig. 6 shows the architecture of the proposed V-CapsNet

Figure 4: Challenging example. Top-1 accuracy is reported only on a validation split of an ImageNet variant in *1807.10108v5.pdf*.

method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43†
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

Table 4. Error rates (%) of **single-model** results on the ImageNet validation set (except † reported on the test set).

Figure 5: Challenging example. The original ResNet paper (*1512.03385.pdf*) only reported on ImageNet validation error rates.

Dataset	Metric	Modality Encoder	Base Encoder	Modified Encoder	OmniVec (Pre.)	OmniVec (FT)
AudioSet(A)	mAP	AST	48.5	49.4	44.7	54.8
AudioSet(A+V)	mAP	AST	-	-	48.6	55.2
SSv2	Top-1 Accuracy	ViViT	65.4	68.6	80.1	85.4
ImageNet1K	Top-1 Accuracy	ViT	88.5	89.1	88.6	92.4
Sun RGBD	Top-1 Accuracy	Simple3D-former	57.3	62.4	71.4	74.6

Table 14. **Impact of increasing backbone size of base modality encoders.** All the base modality encoders above are based on ViT architecture. We increase the number of parameters equivalent to our OmniVec-4 model, by replicating the number of layers.

Figure 6: Challenging example. In *omniVec_2023.pdf*, the ImageNet Top-1 value (92.4 %) appears as one cell in a table containing multiple datasets.

Table 2. Classification accuracies on ImageNet (ResNet-50)

Loss Fn.	Parameter	Top-1	Top-5
CE		76.39	93.20
OHEM	$\rho = 0.8$	76.27	93.21
FL	$\gamma = 0.5$	76.72	93.06
AL (ours)	$\gamma = 0.5$	**76.82**	93.03

Figure 7: Challenging example. *1909.11155v1.pdf* gives Top-1 accuracy only within a table; the metric is absent from the surrounding text.

Method	shapes	Top-1 acc.	Top-5 acc.
baseline [28]	-	51.2%	73.2%
Case 1: α	$\alpha \in \mathbb{R}^{o\times 1\times 1}$	55.5%	78.5%
Case 2: α	$\alpha \in \mathbb{R}^{o\times h_{out}\times w_{out}}$	56.1%	79.0%
Case 3: $\alpha \otimes \beta$	$\alpha \in \mathbb{R}^{o}, \beta \in \mathbb{R}^{w_{out}\times h_{out}}$	56.7%	79.5%
Case 4: $\alpha \otimes \beta \otimes \gamma$	$\alpha \in \mathbb{R}^{o}, \beta \in \mathbb{R}^{w_{out}}$ $\gamma \in \mathbb{R}^{h_{out}}$	**57.1%**	**79.9%**

Table 1: Top-1 and Top-5 classification accuracy using a binarized ResNet-18 on Imagenet for various ways of constructing the scaling factor. α, β, γ are statistically learned via back-propagation. Note that, at test time, all of them can be merged into a single factor, and a single element-wise multiplication is required.

Figure 8: Challenging example. *1909.13863v1.pdf* gives Top-1 accuracy only within a table.

0.4% accuracy drop. Meanwhile, with the compressed model the inference is accelerated by 2.2×. For *AlexNet* with the ImageNet dataset, we achieve 4.9× model compression at the cost that the top-5 accuracy drops slightly from 81.3% to 80%. For *GoogLeNet* with the ImageNet dataset, the proposed method also brings 2.9× reduction of the model parameters

Figure 9: Challenging example. *1807.10119v3.pdf* omits Top-1 accuracy, reporting only Top-5 (80 %).