CNN models comparison and analysis

By Zani Nway Oo, Nu Wai Thet, Phyo Myat Oo

To evaluate the performance of various Convolutional Neural Network (CNN) models, we conducted experiments using a diverse set of images. These included simple, everyday life images, blurry and noisy images, as well as culturally specific images such as those related to Burmese traditions. The goal was to understand how well each model could recognize and classify different types of input under varying conditions.

Initial Observations

In the initial round of testing using simple and clear images, we observed that most models performed reasonably well—except for **InceptionV3**, which consistently misclassified images with unusually high confidence scores. This anomaly was immediately concerning, as it indicated that the model was not just uncertain but confidently wrong.

Upon further investigation and debugging, we discovered that the issue stemmed from the use of a **shared preprocessing function** (`preprocess_input`) across all models. This approach ignored the fact that different architectures require their own specific preprocessing techniques. In the case of InceptionV3, using preprocessing meant for another model led to poor performance.

Correction and Re-evaluation

After updating the preprocessing logic to apply the **appropriate preprocessing function based on each model's architecture**, we re-ran the experiments. As shown in **Table 2**, InceptionV3's performance significantly improved and was now able to classify simple images correctly. With this correction, all models achieved an accuracy in the range of **85% to 96%**, indicating a strong overall performance across the board.

Inference Time Comparison

In terms of prediction speed (inference time), we observed the following average times on simple images:

- **VGGNet16**: ~0.40 seconds (fastest)
- **InceptionV3**: ~0.50 seconds
- **EfficientNet**: ~2.93 seconds (slowest)

These results highlight a trade-off between model complexity and inference time, which is an important consideration for deployment in real-time systems.

Key Insights

- Preprocessing is a **crucial step** in the CNN pipeline. Using the **correct preprocessing method tailored to the specific model** architecture is essential for optimal performance.

- A model's poor performance is not always due to its design—**misconfiguration** during preprocessing or input handling can severely degrade accuracy.

- Inference time varies significantly between models, which may impact the choice of model depending on the application requirements (e.g., speed vs. accuracy).

ResNet50	ResNet50 top1_proba	VGGNet16	VGGNet16 top1_proba	InceptionV	InceptionV3 top1_proba	ConvNeXt	ConvNeXt top1_proba	EfficientNet	EfficientNet top1_proba	label
bald_eagle	1	bald_eagle	1	web_site	1	bald_eagle	0.93	bald_eagle	0.34	bald_eagle
American_cham	0.7	American_chamel	0.64	web_site	1	American_chame	0.61	nail	0.15	chameleon
goldfish	1	goldfish	0.98	sock	0.62	goldfish	0.87	honeycomb	0.53	fish
goldfinch	1	goldfinch	1	web_site	0.9	house_finch	0.76	nail	0.46	goldfinch
goldfish	0.95	goldfish	1	web_site	1	goldfish	0.92	goldfish	0.73	goldfish
hen	0.53	hen	0.5	web_site	1	hen	0.67	cock	0.16	hen
mud_turtle	0.62	terrapin	0.21	paddle	0.81	mud_turtle	0.42	chainlink_fen	0.4	mud_turtle
ostrich	1	ostrich	0.97	web_site	1	ostrich	0.84	ostrich	0.16	ostrich
cock	0.72	hen	0.59	flatworm	0.73	hen	0.56	cock	0.29	rooster
great_white_sh	0.85	great_white_shark	0.97	web_site	1	great_white_sha	0.94	great_white_	0.77	shark
electric_ray	0.8	electric_ray	0.86	web_site	1	stingray	0.72	stingray	0.57	stingray
tench	0.48	tench	0.97	web_site	0.85	tench	0.67	tench	0.47	tench
tiger_shark	0.98	tiger_shark	0.98	web_site	1	tiger_shark	0.88	tiger_shark	0.65	tiger_shark
ambulance	1	ambulance	1	hammer	0.98	ambulance	0.83	ambulance	0.82	ambulance
broom	1	broom	1	stopwatch	1	broom	0.8	broom	0.38	broom
limousine	0.2	minivan	0.45	bow	1	beach_wagon	0.38	cab	0.21	car
cauliflower	1	cauliflower	1	pitcher	1	cauliflower	0.76	cauliflower	0.84	cauliflower
stopwatch	0.57	analog_clock	0.56	web_site	1	analog_clock	0.93	strainer	0.32	clock
warplane	0.37	airship	0.26	web_site	1	ambulance	0.63	warplane	0.59	helicopter
safe	0.52	projector	0.14	web_site	1	notebook	0.23	switch	0.12	laptop_with_notebool
speedboat	0.99	speedboat	0.96	web_site	1	speedboat	0.63	speedboat	0.45	speedboat
sunglass	0.16	sunglasses	0.11	clog	1	sunglass	0.6	sunglass	0.21	sunglasses
violin	0.8	violin	0.93	clog	1	violin	0.91	hook	0.05	violin
(Accuracy)		(Accuracy)		(Accuracy)		(Accuracy)		(Accuracy)		
0.87		0.91		0		0.96		0.74		

Table 1: Model comparison on simple images before correcting preprocessing image

ResNet50	ResNet50 top1_proba	ResNet50 time_sec	VGGNet16	VGGNet16 top1_proba	VGGNet16 time_sec	InceptionV	InceptionV3 top1_proba	InceptionV3 time_sec	ConvNeXt	ConvNeXt top1_proba	ConvNeXt time_sec	EfficientN	EfficientNet top1_proba	EfficientNet time_sec	label
bald_eagle	1	2.69	bald_eagle	1	0.68	bald_eagle	0.93	3.15	bald_eagl	0.93	3.36	bald_eagl	0.8	10.86	bald_eagle
American_(	0.7	0.27	American_cha	0.64	0.4	American_(	0.71	0.26	American_	0.61	0.88	American_	0.51	2.12	chameleon
goldfish	1	0.2	goldfish	0.98	0.36	goldfish	0.99	0.34	goldfish	0.87	0.95	goldfish	0.81	2.12	fish
goldfinch	1	0.27	goldfinch	1	0.32	goldfinch	0.96	0.25	house_fin	0.76	0.94	goldfinch	0.82	2.22	goldfinch
goldfish	0.95	0.2	goldfish	1	0.37	goldfish	0.99	0.23	goldfish	0.92	0.97	goldfish	0.83	2.19	goldfish
hen	0.53	0.27	hen	0.5	0.43	hen	0.62	0.21	hen	0.67	0.98	hen	0.8	2.16	hen
mud_turtle	0.62	0.35	terrapin	0.21	0.37	mud_turtle	0.68	0.29	mud_turtl	0.42	0.96	mud_turtl	0.73	2.15	mud_turtle
ostrich	1	0.27	ostrich	0.97	0.4	ostrich	1	0.36	ostrich	0.84	0.89	ostrich	0.8	2.13	ostrich
cock	0.72	0.34	hen	0.59	0.33	hen	0.59	0.31	hen	0.56	0.95	hen	0.66	2.29	rooster
great_whit	0.85	0.2	great_white_s	0.97	0.41	great_whit	0.66	0.22	great_whi	0.94	0.92	great_whi	0.85	2.2	shark
electric_ray	0.8	0.23	electric_ray	0.86	0.35	stingray	0.5	0.21	stingray	0.72	1.16	stingray	0.71	2.92	stingray
tench	0.48	0.29	tench	0.97	0.41	tench	0.92	0.31	tench	0.67	1.4	tench	0.85	2.89	tench
tiger_shark	0.98	0.24	tiger_shark	0.98	0.46	tiger_shark	0.95	0.28	tiger_shar	0.88	1.16	tiger_shar	0.81	2.48	tiger_shark
ambulance	1	3.74	ambulance	1	0.52	ambulance	0.95	3.43	ambulanc	0.83	2.67	ambulanc	0.83	10.97	ambulance
bow_tie	1	0.22	bow_tie	0.95	0.37	bow_tie	1	0.21	bow_tie	0.9	0.9	bow_tie	0.75	2.12	bow
broom	1	0.33	broom	1	0.35	broom	0.76	0.23	broom	0.8	1.05	broom	0.81	2.15	broom
limousine	0.2	0.2	minivan	0.45	0.39	sports_car	0.22	0.27	beach_wa	0.38	0.85	minivan	0.23	2.13	car
cauliflower	1	0.25	cauliflower	1	0.38	cauliflower	0.97	0.28	cauliflowe	0.76	0.82	cauliflowe	0.81	2.1	cauliflower
stopwatch	0.57	0.42	analog_clock	0.56	0.45	stopwatch	0.78	0.42	analog_cl	0.93	0.87	analog_cl	0.83	2.18	clock
clog	0.99	0.32	clog	0.88	0.35	clog	0.71	0.22	clog	0.83	0.85	clog	0.82	2.15	clog
hammer	0.6	0.72	hammer	0.71	0.36	hammer	0.94	0.34	hammer	0.69	0.91	nail	0.48	2.19	hammer
warplane	0.37	0.36	airship	0.26	0.34	ambulance	0.4	0.24	ambulanc	0.63	0.98	ambulanc	0.32	2.19	helicopter
safe	0.52	0.3	projector	0.14	0.33	nail	0.04	0.33	notebook	0.23	1	notebook	0.63	2.42	laptop with noteboo
pitcher	0.52	0.29	pitcher	0.73	0.41	pitcher	0.3	0.22	pitcher	0.43	0.83	pitcher	0.36	2.14	pitcher
speedboat	0.99	0.31	speedboat	0.96	0.33	speedboat	0.75	0.28	speedboa	0.63	0.92	speedboa	0.38	2.19	speedboat
sunglass	0.16	0.23	sunglasses	0.11	0.39	sunglasses	0.54	0.26	sunglass	0.6	1.02	sunglass	0.41	2.54	sunglasses
violin	0.8	0.35	violin	0.93	0.52	violin	0.97	0.38	violin	0.91	1.42	violin	0.74	2.91	violin
(Accuracy)		avg time(s	(Accuracy)		avg time(s)	(Accuracy)		avg time(s)	(Accuracy)		avg time(s)	(Accuracy)		avg time(s)	
0.85		0.51	0.89		0.4	0.93		0.5	0.96		1.13	0.93		2.93	

Table 2: Model comparison on simple images after correcting preprocessing image

```python
def classify_image(self, name, img, top_k=1):
    model = self.get_model(name)
    img = img.resize((model.input_shape[1], model.input_shape[2]))
    x = img_to_array(img)
    x = np.expand_dims(x, axis=0)

    # Use correct preprocessing
    if name == 'InceptionV3':
        x = inception_preprocess(x)
    elif name == 'ResNet50':
        x = resnet_preprocess(x)
    elif name == 'VGGNet16':
        x = vgg_preprocess(x)
    elif name == 'EfficientNet':
        x = effnet_preprocess(x)
    else:
        x = preprocess_input(x)  # default if unknown

    preds = model.predict(x, verbose=0)
    return decode_predictions(preds, top=top_k)
```

Fig1 : Correct preprocessing Steps

CNN Model Evaluation on Noisy, Blurry & Confusing Images

Objective

We aimed to test the robustness of CNN models (ResNet50, InceptionV3, VGGNet16, EfficientNet, ConvNeXt) on challenging image inputs—blurry, noisy, low-quality, or visually confusing images. For each image, we considered the **top-3 predicted classes and probabilities** to assess how well models can still offer relevant guesses, even when the top-1 prediction may not be fully accurate.

Key Observations & Findings

1. Top-3 Prediction Helps Clarify Ambiguities

In most cases, even when the top-1 class was not perfectly aligned with the label (especially due to manual labeling differences), the correct category appeared in the **top-3 predictions**. For example:

- Images labeled "**dog wearing sunglasses**" were predicted as *sunglasses*, *sunglass*, and relevant dog breeds across all models.

- "**Wine glass with flower**" matched with *goblet*, *red wine*, or *vase*, which shows **semantic proximity** is preserved.

2. Misclassifications Often Reasonable

The incorrect guesses often reflect **visually similar or related classes**:

- *Camouflage_gun* images predicted as *screw*, *hook*, or *revolver*.

- *Motorbike-related* images guessed as *mountain_bike*, *breakwater*, *wing*, *bubble*, etc., due to abstract or partial visuals.

This behavior suggests the models **attempt to generalize** based on visible patterns even when **clarity is poor**.

3. Model Comparison on Challenging Inputs

Model	Observations
ResNet50	Correct top-1 predictions for 15/33 cases. Strong top-3 semantic matches, though sometimes confidently wrong on noisy inputs (e.g., *mouse* for a cap). Strong top-3 semantic matches; slightly overconfident on irrelevant classes in noisy contexts (e.g., predicting *mouse* for a cap).
InceptionV3	Achieved 19/33 correct top-1 results. Confident but occasionally "hallucinates" on visually complex or noisy images (e.g., *alp*, *warplane*).
VGGNet16	Correct top-1 in 15/33 cases. Conservative in predictions; often produces reasonable alternatives (e.g., *hook*, *nail*, *screw* for tool-like shapes).
EfficientNet	Highest top-1 accuracy (20/33). Shows robustness under noisy conditions; classifies fine-grained categories (e.g., *cat breeds*, *pizza types*) well. top-3 often includes class clusters (e.g., *tabby*, *Egyptian_cat*, *tiger_cat* for cat).
ConvNeXt	17/33 top-1 correct. Strong semantic grouping in top-3 even when top-1 is off; good resilience in ambiguous contexts (e.g., *cat*, *dog*, *toy*). stable in semantic grouping within top-3. Often included all relevant options even when confidence was low.

Challenging Input Example: Scissors Misclassified by All Models

Fig2: All five CNN models misclassified this image of a pair of scissors.

The predictions varied widely: ResNet50 predicted *pill bottle*, VGGNet16 suggested *syringe*, InceptionV3 identified a *loupe*, ConvNeXt leaned toward *nipple*, and EfficientNet returned *hook*.

Despite being incorrect, several predictions show the models were associating tool-like shapes. This highlights the challenge of visual ambiguity and the importance of top-3 evaluation in edge cases.

4. Domain Gap Analysis: Out-of-Distribution Performance

To evaluate model robustness beyond the ImageNet domain, we tested models on a culturally-specific dataset containing 21 images of Myanmar cultural artifacts and architecture. These visual concepts were chosen because they are unlikely to appear in the ImageNet-1K taxonomy https://deeplearning.cms.waikato.ac.nz/user-guide/class-maps/IMAGENET/.

Our hypothesis:

- ImageNet-pretrained models would struggle with out-of-distribution (OOD) data.
- Cultural content from Myanmar would highlight limitations in generalization, especially given the Western-centric nature of common training datasets.

Key findings:

- Experimental results confirmed poor generalization to culturally-specific content.
- To better understand model uncertainty, we analyzed Top-3 prediction probabilities.
- Confidence distribution box plot revealed how secondary and tertiary predictions behave, offering deeper insights into model uncertainty.

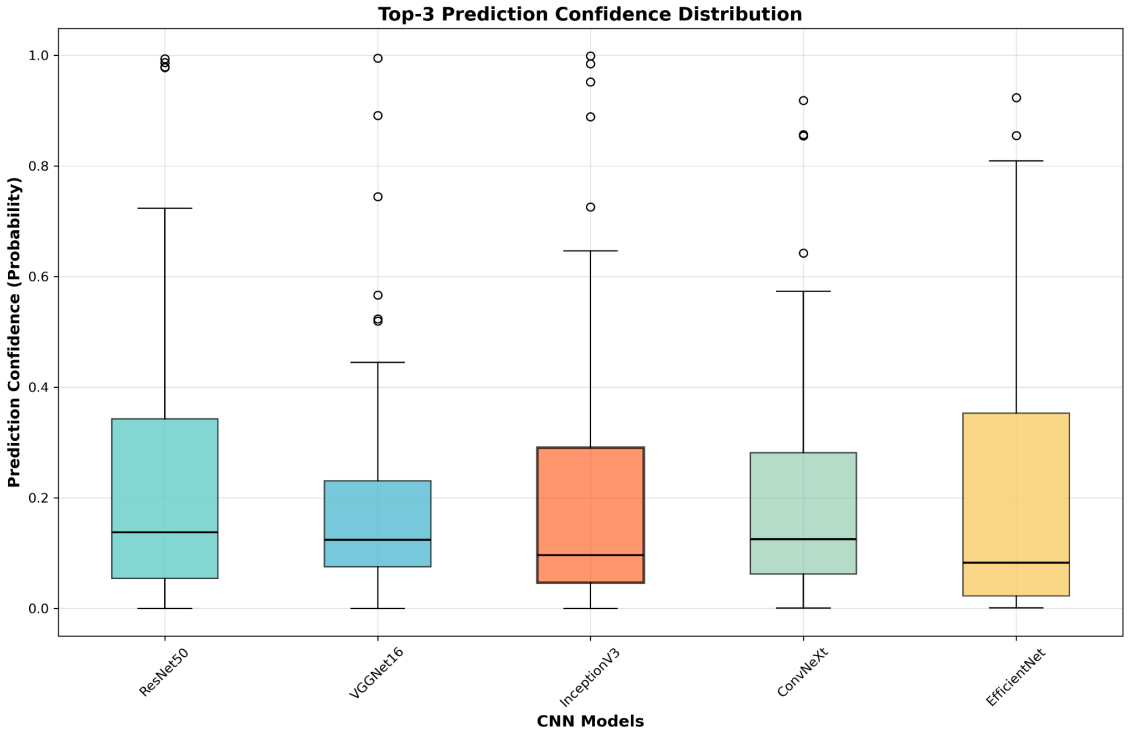

Fig 3: Top-3 Prediction Confidence Distribution

Model	Median	Q1	Q3	IQR
ResNet50	0.138	0.053	0.360	0.307
ConvNeXt	0.125	0.061	0.301	0.241
VGGNet16	0.124	0.074	0.231	0.158
InceptionV3	**0.096**	**0.046**	**0.301**	**0.254**
EfficientNet	0.083	0.021	0.361	0.339

Table 3: Model Confidence Distribution Results of All Top-3 Predictions

The boxplot analysis shows that all five CNN models exhibit low confidence on Myanmar cultural imagery.

- Top-3 predictions: Median confidence scores remain low (**0.083** to **0.138**), highlighting difficulty with this out-of-domain content.
- Top-1 predictions: Average Confidence ranges between **38.4%** and **55.0%**, overall uncertainty remains high.

This pattern reflects typical model behavior when facing visual concepts far removed from their training distribution.

Model	Avg. Confidence %	Avg. time (s)	High Confidence %	Overall Score (Normalized)
ResNet50	55%	0.066	19.0%	95.7
InceptionV3	**50.7%**	**0.061**	**19.0%**	**86.9**
EfficientNet	53.9%	0.399	14.3%	46.8
VGGNet16	38.4%	0.08	9.5%	36.2
ConvNeXt	46.3%	0.297	19.0%	26.8

Table 4: Normalized Evaluation Framework for Top-1 Predictions of Models

Our evaluation combines prediction confidence and inference speed using a normalized 0–100 scoring system, providing a balanced comparison across models.
Key results:

- Top performers: **ResNet50 (95.7)** and **InceptionV3 (86.9)** lead with strong confidence and fast inference.

- High-confidence predictions (>80%): ResNet50, InceptionV3, and ConvNeXt achieve **19.0%**, while VGGNet16 and EfficientNet lag behind at **9.5%** and **14.3%**.

These results highlight clear differences in reliability across architectures when dealing with out-of-distribution cultural imagery.

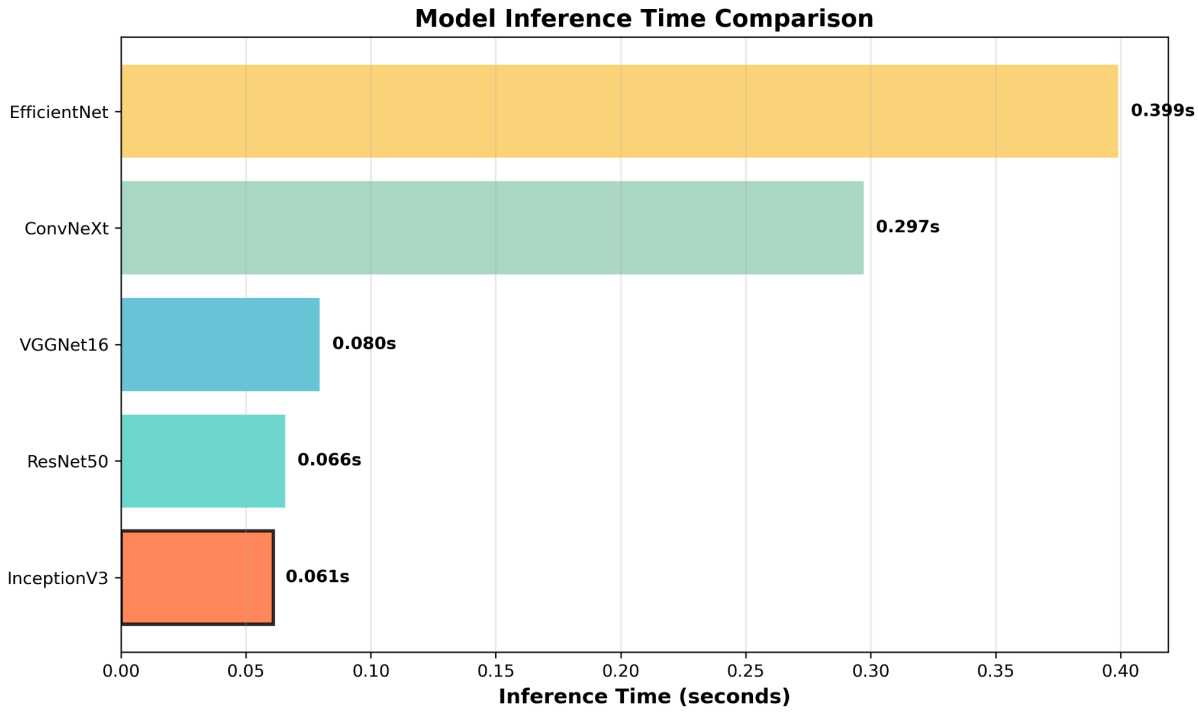

Fig 4: Model Inference Time Comparison

In terms of inference speed, **InceptionV3** is the fastest at **0.061** seconds per image, followed closely by ResNet50 at **0.066** seconds. InceptionV3's efficient design, using inception modules, offers a slight speed advantage without sacrificing performance. Both models show excellent real-time suitability where low latency is essential.

Detailed Analysis of the Predicted Classifications and Ground Truth

Pagoda

Understanding cultural context often starts with religion and its architectural forms, as they reflect a nation's identity. To explore this, I tested the models on four well-known religious pagodas from Myanmar.

Key points:

- Religious architecture captures deep cultural identity.
- Myanmar pagodas are unlikely to be well-represented in ImageNet.
- With a small test set, we aimed for reasonable proximity, not exact matches.

- Google Image Search was used to visualize and assess model outputs, focusing on the most visually similar results from the first page.

label	ResNet50_top1	ResNet50_top2	ResNet50_top3
pagoda_kyitehteeyoe	obelisk	megalith	pedestal
pagoda_myatheintan	stupa	palace	mosque
pagoda_shwedagon	stupa	palace	fountain
pagoda_suelay	stupa	palace	monastery
label	**VGGNet16_top1**	**VGGNet16_top2**	**VGGNet16_top3**
pagoda_kyitehteeyoe	lemon	obelisk	plastic_bag
pagoda_myatheintan	stupa	palace	mosque
pagoda_shwedagon	stupa	palace	castle
pagoda_suelay	stupa	palace	mosque
label	**InceptionV3_top1**	**InceptionV3_top2**	**InceptionV3_top3**
pagoda_kyitehteeyoe	punching_bag	hook	plastic_bag
pagoda_myatheintan	palace	stupa	mosque
pagoda_shwedagon	stupa	liner	Indian_elephant
pagoda_suelay	stupa	palace	liner
label	**ConvNeXt_top1**	**ConvNeXt_top2**	**ConvNeXt_top3**
pagoda_kyitehteeyoe	megalith	dome	church
pagoda_myatheintan	stupa	palace	mosque
pagoda_shwedagon	stupa	palace	dome
pagoda_suelay	stupa	palace	mosque
label	**EfficientNet_top1**	**EfficientNet_top2**	**EfficientNet_top3**
pagoda_kyitehteeyoe	stupa	church	monastery
pagoda_myatheintan	stupa	palace	mosque
pagoda_shwedagon	stupa	palace	monastery
pagoda_suelay	stupa	palace	mosque

Table 5: Model Classifications for Different Myanmar Pagoda Images

Fig 5: pagoda_myatheintan input and visualization of outputs

The model's output for this pagoda showed strong links to Indian culture, which aligns with the historical influence of Indian Buddhist architecture on Myanmar.

Key points:

- Results included Indian cultural elements
- Other results like **'stupa'** and **'mosque'** suggest ImageNet is more exposed to widely recognized Asian cultural concepts, especially from India and China.
- While exact matches were rare, the model likely recognized culturally similar patterns.
- In Myanmar, these structures are called **'pagodas'**, but the dataset seems to associate them more with the term **'stupa'**.

Fig 6: pagoda_kyitehteeyoe input and visualization of outputs

The model's predictions for this image were often out of context, revealing clear challenges with culturally specific, underrepresented content.

Key points:

- Bizarre predictions included **"punching bag", "plastic bag", "hook"**, and **"pedestal".**
- Reasonable guesses: **"stupa," "obelisk," and "megalith"** — possibly influenced by the large blue sky background, common in search results for these terms.

- **“Lemon”** appeared as a surprising top-1 prediction, likely due to the golden, rounded shape of the structure.
- Two models even predicted **“punching bag”** as top-1, which was unexpected.
- These results highlight how CNNs, when limited to pre-trained datasets, struggle with culturally specific and underrepresented categories, even when the objects are obvious to humans.

Fig 7: pagoda_shwedagon input and visualization of outputs

The same issue appears with Top-2 and Top-3 predictions, showing strange and unrelated results.

Key points:

- Odd predictions included **“Indian elephant”** and **“liner”**, despite no visual resemblance to an elephant.
- Likely influenced by the large blue sky background, similar to the previous example.
- Highlights how irrelevant background elements can confuse pre-trained models when handling unfamiliar, culturally specific imagery.

Fig 8: pagoda_suelay input and visualization of outputs

Aside from the **"liner"** prediction, most of the model outputs generally make sense in this context. However, the **"monastery"** prediction is also a bit unusual, as the monasteries I found through Google Search do not closely resemble the input image. That being said, since we do not have direct access to the actual ImageNet dataset, we have to work with the available results.

From these observations, we can see that the model often detects structurally similar forms even when predicted keywords don't match expected terms.

- This may stem from naming differences or a lack of Myanmar-specific examples in training data.
- While visual patterns are recognized, the predictions remain inaccurate.
- In religious architecture, subtle design differences hold important cultural meaning.
- Mistaking one structure for another, despite shape similarities, leads to misleading classifications.

Monk

label	**ResNet50_top1**	**ResNet50_top2**	**ResNet50_top3**
monk	umbrella	gown	kimono
label	**VGGNet16_top1**	**VGGNet16_top2**	**VGGNet16_top3**
monk	umbrella	broom	quill
label	**InceptionV3_top1**	**InceptionV3_top2**	**InceptionV3_top3**
monk	umbrella	abaya	kimono
label	**ConvNeXt_top1**	**ConvNeXt_top2**	**ConvNeXt_top3**
monk	cellular_telephone	notebook	umbrella
label	**EfficientNet_top1**	**EfficientNet_top2**	**EfficientNet_top3**
monk	umbrella	notebook	Indian_cobra

Table 6: Model Classifications for Myanmar Monk Image

Fig 9: monk input and visualization of outputs

After testing on religious architecture, we evaluated a Myanmar young monk image containing three elements: the monk holding a religious book and a red umbrella.

Key points:

- Models mostly focused on the umbrella, with top-1 predictions often being **“umbrella”**.
- ConvNeXT surprisingly predicted **“cellular telephone”** as top-1, despite no phone-like object in the image.
- Top-2 and top-3 predictions are sometimes related to **clothing** or a **notebook**, which fits the context.
- However, some predictions like **“broom,” “quill,”** and **“Indian cobra”** were confusing.

Insights & Summary

- **Top-3 predictions offer a broader perspective** when evaluating model performance, particularly in challenging or ambiguous cases. This approach accounts for the possibility that multiple labels may be visually or semantically plausible—especially important when dealing with non-standard or out-of-distribution (non-ImageNet) images.

- **EfficientNet and Inception show superior semantic resilience**, maintaining consistent and contextually appropriate predictions even under visual distortions. These models were less likely to produce erratic outputs and often included the correct or related class within the top-3 results.

- **Misclassifications often stem from label ambiguity** rather than complete failure. For example, artistic renderings, close-up views, or partially occluded objects can lead to confusion. However, the models still tend to predict conceptually related categories, revealing that their internal feature representations remain strong.

- **Visual confusion**—caused by low-level issues like shadows, object blending, or unusual angles—can produce outlier predictions (e.g., mistaking a hammer for a beer glass). This suggests current CNNs may still struggle with contextual inference at the pixel level.

- **Top-1 accuracy alone is insufficient** for complex visual scenarios. A top-3 evaluation strategy provides a more complete picture of model understanding, especially when human interpretations may not align perfectly with the predefined labels in the training set.

This analysis confirms that **top-1 prediction alone can be misleading in complex visual contexts**, and evaluating **top-3 predictions gives a more holistic view** of model understanding—especially when human-labeled ground truths may differ from standard class labels used in pretrained models.

Conclusion

Across all five models, we observed a consistent trend: **CNN architectures retain semantic awareness under distortion**. While top-1 predictions may sometimes fail in ambiguous cases, top-3 predictions often include correct or closely related classes, reflecting the models' ability to generalize meaningfully.

This confirms that in real-world applications—where input quality and label clarity vary—**evaluating beyond top-1 accuracy is essential**. Robustness in top-3 predictions indicates that even when models "fail" by strict classification standards, they often still preserve useful, interpretable representations of the input.

To research more on the Inception architecture, its building blocks are built from scratch.

- https://www.kaggle.com/code/phyomyatoo/inception-from-version-1-on-cifar-10