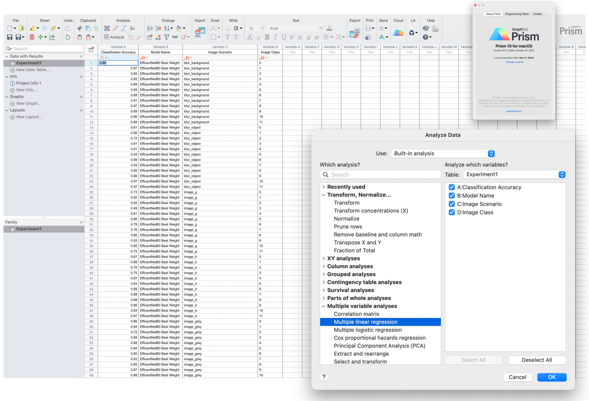
XIMAGENET-12: An Explainable Visual Benchmark Dataset for Model Robustness Evaluation

Supplementary Material

7. Appendix

In this section we provide the supplementary compiled together with the main paper includes:

- The illustration of how we use Multiple Linear Regression to verify our hypothesis: from raw data input, for example, in GraphPad Prism, to interpreting examples and residual plots, etc;
- The training details (Accuracy and Loss Plots) and hyperparameters within scenarios and cross-scenario experiments, including diffusion metrics for evaluations, density maps of State-of-the-Art accuracy drop (e.g., referring to our particular experiment, EX1, EX2);
- The ablation study addresses the industry pain points, illustrating robust model selection for challenging scenarios, particularly due to factors such as background variations, camera shifts, color changes, and lighting conditions;
- Sample image of our XIMAGENEt-12 AI-generated image, comprising 12,248 images for AI-generated scenarios using the latest Stable Diffusion XL model and involving uniform promotion and manual selection and filtering.



	Classification Accuracy	Model Name	Image Scenario	Image Class
1	0.89	EfficentNetB0 Best Weight	blur_background	0
2	0.97	EfficentNetB0 Best Weight	blur_background	1
3	0.85	EfficentNetB0 Best Weight	blur_background	2
4	0.92	EfficentNetB0 Best Weight	blur_background	3
5	0.89	EfficentNetB0 Best Weight	blur_background	4
6	0.95	EfficentNetB0 Best Weight	blur_background	5
7	0.88	EfficentNetB0 Best Weight	blur_background	6
8	0.91	EfficentNetB0 Best Weight	blur_background	7
9	0.97	EfficentNetB0 Best Weight	blur_background	8
10	0.89	EfficentNetB0 Best Weight	blur_background	9
11	0.96	EfficentNetB0 Best Weight	blur_background	10
12	0.89	EfficentNetB0 Best Weight	blur_background	11
13	0.81	EfficentNetB0 Best Weight	blur_object	0
14	0.96	EfficentNetB0 Best Weight	blur_object	1
15	0.72	EfficentNetB0 Best Weight	blur_object	2
16	0.91	EfficentNetB0 Best Weight	blur_object	3
17	0.91	EfficentNetB0 Best Weight	blur_object	4
18	0.93	EfficentNetB0 Best Weight	blur_object	5
19	0.89	EfficentNetB0 Best Weight	blur_object	6
20	0.93	EfficentNetB0 Best Weight	blur_object	7
21	0.95	EfficentNetB0 Best Weight	blur_object	8
22	0.96	EfficentNetB0 Best Weight	blur_object	9
23	0.97	EfficentNetB0 Best Weight	blur_object	10
24	0.97	EfficentNetB0 Best Weight	blur_object	11
25	0.73	EfficentNetB0 Best Weight	image_g	0
26	0.92	EfficentNetB0 Best Weight	image_g	1
27	0.62	EfficentNetB0 Best Weight	image_g	2
28	0.49	EfficentNetB0 Best Weight	image_g	3
29	0.81	EfficentNetB0 Best Weight	image_g	4
30	0.86	EfficentNetB0 Best Weight	image_g	5
31	0.79	EfficentNetB0 Best Weight	image_g	6
32	0.76	EfficentNetB0 Best Weight	image_g	7
33	0.86	EfficentNetB0 Best Weight	image_g	8
34	0.93	EfficentNetB0 Best Weight	image_g	9
35	0.85	EfficentNetB0 Best Weight	image_g	10
36	0.75	EfficentNetB0 Best Weight	image_g	11
37	0.67	EfficentNetB0 Best Weight	image_b	0
38	0.88	EfficentNetB0 Best Weight	image_b	1
39	0.70	EfficentNetB0 Best Weight	image_b	2
40	0.75	EfficentNetB0 Best Weight	image_b	3
41	0.87	EfficentNetB0 Best Weight	image_b	4
42	0.93	EfficentNetB0 Best Weight	image_b	5
43	0.89	EfficentNetB0 Best Weight	image_b	6
44	0.88	EfficentNetB0 Best Weight	image_b	7
45	0.88	EfficentNetB0 Best Weight	image_b	8
46	0.66	EfficentNetB0 Best Weight	image_b	9
47	0.83	EfficentNetB0 Best Weight	image_b	10
48	0.97	EfficentNetB0 Best Weight	image_b	11
49	0.86	EfficentNetB0 Best Weight	image_grey	0
50	0.94	EfficentNetB0 Best Weight	image_grey	1
51	0.72	EfficentNetB0 Best Weight	image_grey	2
52	0.68	EfficentNetB0 Best Weight	image_grey	3
53	0.88	EfficentNetB0 Best Weight	image_grey	4
54	0.94	EfficentNetB0 Best Weight	image_grey	5
55	0.90	EfficentNetB0 Best Weight	image_grey	6
56	0.93	EfficentNetB0 Best Weight	image_grey	7
57	0.85	EfficentNetB0 Best Weight	image_grey	8
58	0.87	EfficentNetB0 Best Weight	image_grey	9
59	0.88	EfficentNetB0 Best Weight	image_grey	10

Analyze Data

Use: Built-in analysis

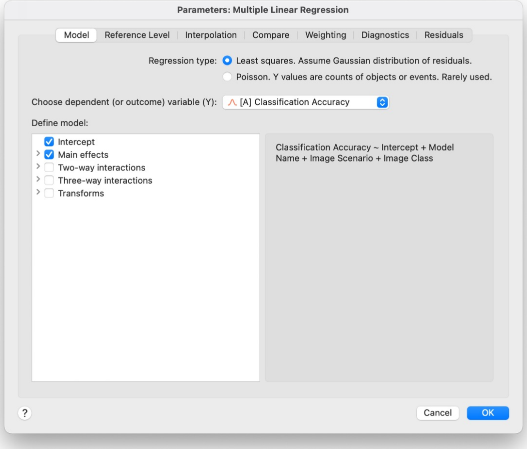
Which analysis?

- Recently used
- Transform, Normalize...
  - Transform
  - Transform concentrations (X)
  - Normalize
  - Prune rows
  - Remove baseline and column math
  - Transpose X and Y
  - Fraction of Total
- XY analyses
- Column analyses
- Grouped analyses
- Contingency table analyses
- Survival analyses
- Parts of whole analyses
- Multiple variable analyses
  - Correlation matrix
  - Multiple linear regression
  - Multiple logistic regression
  - Cox proportional hazards regression
  - Principal Component Analysis (PCA)
  - Extract and rearrange
  - Select and transform

Analyze which variables? Table: Experiment1

☑ A:Classification Accuracy
☑ B:Model Name
☑ C:Image Scenario
☑ D:Image Class

Define the raw data type and variable into statistic software (GraphPad Prism)



Parameters: Multiple Linear Regression

Regression type: ● Least squares. Assume Gaussian distribution of residuals.
○ Poisson. Y values are counts of objects or events. Rarely used.

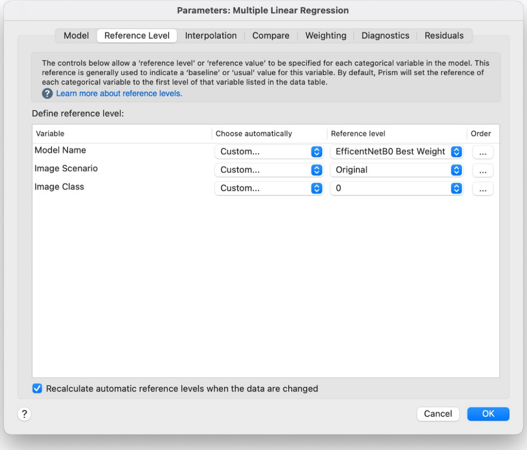
Choose dependent (or outcome) variable (Y): [A] Classification Accuracy

Define model:

☑ Intercept
☑ Main effects
□ Two-way interactions
□ Three-way interactions
□ Transforms

Classification Accuracy ~ Intercept + Model Name + Image Scenario + Image Class

Choose Regression type and define the base independent variables (Model, Image Scenario, Image Class)



Parameters: Multiple Linear Regression

The controls below allow a 'reference level' or 'reference value' to be specified for each categorical variable in the model. This reference is generally used to indicate a 'baseline' or 'usual' value for this variable. By default, Prism will set the reference of each categorical variable to the first level of that variable listed in the data table.

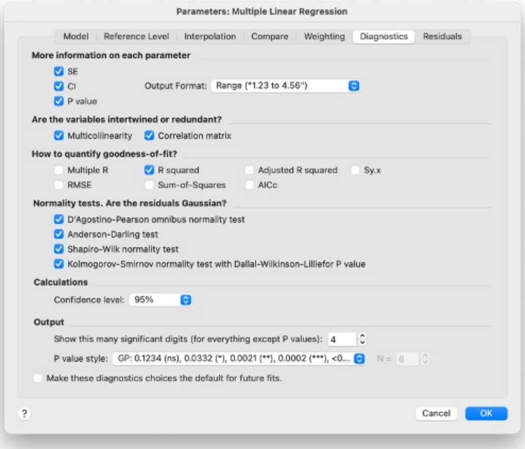
Learn more about reference levels.

Define reference level:

Variable	Choose automatically	Reference level	Order
Model Name	Custom...	EfficentNetB0 Best Weight	...
Image Scenario	Custom...	Original	...
Image Class	Custom...	0	...

☑ Recalculate automatic reference levels when the data are changed

Select reference level for each independent variables (Model, Image Scenario, Image Class)



Parameters: Multiple Linear Regression

More information on each parameter

☑ SE
☑ CI Output Format: Range ("1.23 to 4.56")
☑ P value

Are the variables intertwined or redundant?

☑ Multicollinearity ☑ Correlation matrix

How to quantify goodness-of-fit?

□ Multiple R ☑ R squared □ Adjusted R squared □ Sy.x
□ RMSE □ Sum-of-Squares □ AICc

Normality tests. Are the residuals Gaussian?

☑ D'Agostino-Pearson omnibus normality test
☑ Anderson-Darling test
☑ Shapiro-Wilk normality test
☑ Kolmogorov-Smirnov normality test with Dallal-Wilkinson-Lilliefor P value

Calculations

Confidence level: 95%

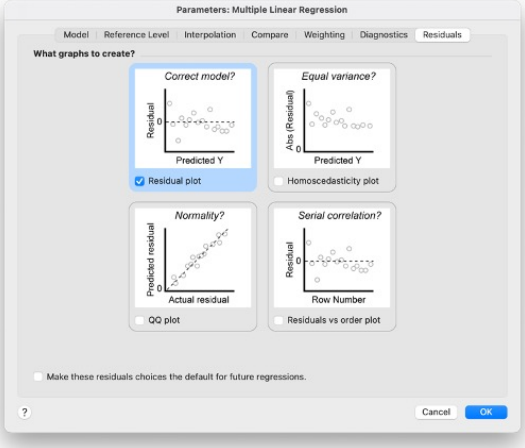
Output

Show this many significant digits (for everything except P values): 4

P value style: GP: 0.1234 (ns), 0.0332 (*), 0.0021 (**), 0.0002 (***), <0... N = 6

□ Make these diagnostics choices the default for future fits.

Set Parameters for Multiple Liner Regression, such as Confidence Level etc



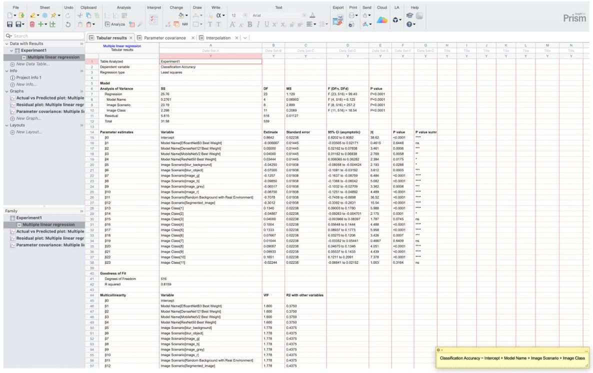
Parameters: Multiple Linear Regression

What graphs to create?

☑ Residual plot
□ Homoscedasticity plot
□ QQ plot
□ Residuals vs order plot

□ Make these residuals choices the default for future regressions.

Create target residual plot graph for simulating the regression results



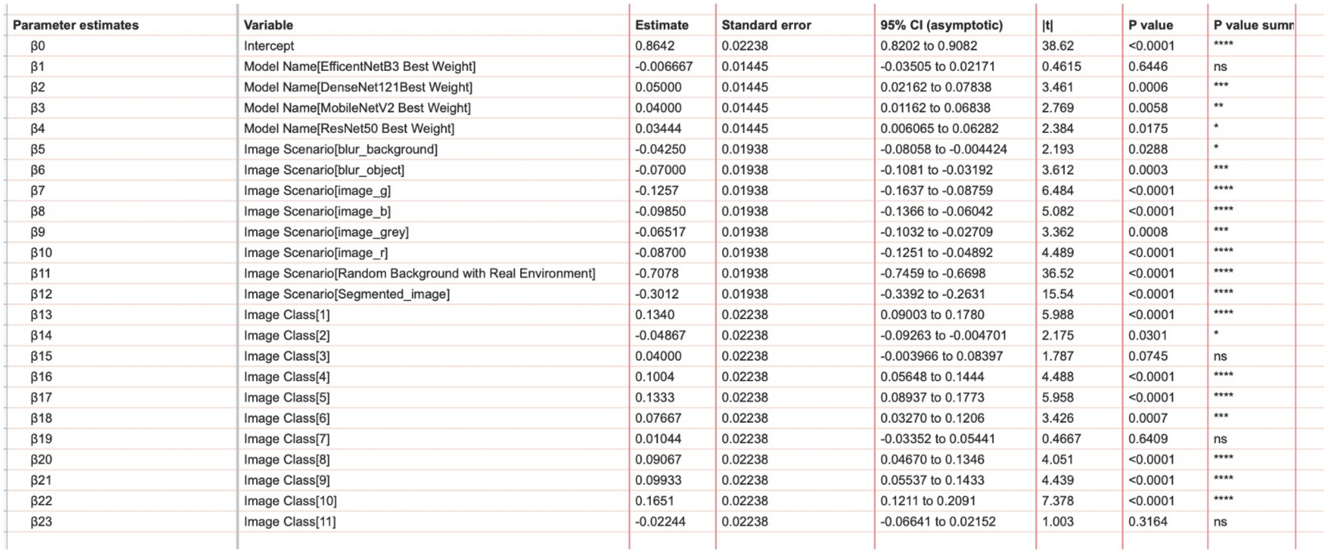
Multiple linear regression
Tabular results

		A	B	C	D	E	F	G
1	Table Analyzed	Experiment1						
2	Dependent variable	Classification Accuracy						
3	Regression type	Least squares						
4								
5	**Model**							
6	**Analysis of Variance**	**SS**	**DF**	**MS**	**F (DFn, DFd)**	**P value**		
7	Regression	25.76	23	1.120	F (23, 516) = 99.40	P<0.0001		
8	Model Name	0.2761	4	0.06903	F (4, 516) = 6.125	P<0.0001		
9	Image Scenario	23.19	8	2.899	F (8, 516) = 257.2	P<0.0001		
10	Image Class	2.298	11	0.2089	F (11, 516) = 18.54	P<0.0001		
11	Residual	5.815	516	0.01127				
12	Total	31.58	539					
13								
14	**Parameter estimates**	**Variable**	**Estimate**	**Standard error**	**95% CI (asymptotic)**	**|t|**	**P value**	**P value sumr**
15	β0	Intercept	0.8642	0.02238	0.8202 to 0.9082	38.62	<0.0001	****
16	β1	Model Name[EfficentNetB3 Best Weight]	-0.006667	0.01445	-0.03505 to 0.02171	0.4615	0.6446	ns
17	β2	Model Name[DenseNet121Best Weight]	0.05000	0.01445	0.02162 to 0.07838	3.461	0.0006	***
18	β3	Model Name[MobileNetV2 Best Weight]	0.04000	0.01445	0.01162 to 0.06838	2.769	0.0058	**
19	β4	Model Name[ResNet50 Best Weight]	0.03444	0.01445	0.006065 to 0.06282	2.384	0.0175	*
20	β5	Image Scenario[blur_background]	-0.04250	0.01938	-0.08058 to -0.004424	2.193	0.0288	*
21	β6	Image Scenario[blur_object]	-0.07000	0.01938	-0.1081 to -0.03192	3.612	0.0003	***
22	β7	Image Scenario[image_g]	-0.1257	0.01938	-0.1637 to -0.08759	6.484	<0.0001	****
23	β8	Image Scenario[image_b]	-0.09850	0.01938	-0.1366 to -0.06042	5.082	<0.0001	****
24	β9	Image Scenario[image_grey]	-0.06517	0.01938	-0.1032 to -0.02709	3.362	0.0008	***
25	β10	Image Scenario[image_r]	-0.08700	0.01938	-0.1251 to -0.04892	4.489	<0.0001	****
26	β11	Image Scenario[Random Background with Real Environment]	-0.7078	0.01938	-0.7459 to -0.6698	36.52	<0.0001	****
27	β12	Image Scenario[Segmented_image]	-0.3012	0.01938	-0.3392 to -0.2631	15.54	<0.0001	****
28	β13	Image Class[1]	0.1340	0.02238	0.09003 to 0.1780	5.988	<0.0001	****
29	β14	Image Class[2]	-0.04867	0.02238	-0.09263 to -0.004701	2.175	0.0301	*
30	β15	Image Class[3]	0.04000	0.02238	-0.003966 to 0.08397	1.787	0.0745	ns
31	β16	Image Class[4]	0.1004	0.02238	0.05648 to 0.1444	4.488	<0.0001	****
32	β17	Image Class[5]	0.1333	0.02238	0.08937 to 0.1773	5.958	<0.0001	****
33	β18	Image Class[6]	0.07667	0.02238	0.03270 to 0.1206	3.426	0.0007	***
34	β19	Image Class[7]	0.01044	0.02238	-0.03352 to 0.05441	0.4667	0.6409	ns
35	β20	Image Class[8]	0.09067	0.02238	0.04670 to 0.1346	4.051	<0.0001	****
36	β21	Image Class[9]	0.09933	0.02238	0.05537 to 0.1433	4.439	<0.0001	****
37	β22	Image Class[10]	0.1651	0.02238	0.1211 to 0.2091	7.378	<0.0001	****
38	β23	Image Class[11]	-0.02244	0.02238	-0.06641 to 0.02152	1.003	0.3164	ns
39								
40	**Goodness of Fit**							
41	Degrees of Freedom	516						
42	R squared	0.8159						
43								
44	**Multicollinearity**	**Variable**	**VIF**	**R2 with other variables**				
45	β0	Intercept						
46	β1	Model Name[EfficentNetB3 Best Weight]	1.600	0.3750				
47	β2	Model Name[DenseNet121Best Weight]	1.600	0.3750				
48	β3	Model Name[MobileNetV2 Best Weight]	1.600	0.3750				
49	β4	Model Name[ResNet50 Best Weight]	1.600	0.3750				
50	β5	Image Scenario[blur_background]	1.778	0.4375				
51	β6	Image Scenario[blur_object]	1.778	0.4375				
52	β7	Image Scenario[image_g]	1.778	0.4375				
53	β8	Image Scenario[image_b]	1.778	0.4375				
54	β9	Image Scenario[image_grey]	1.778	0.4375				
55	β10	Image Scenario[image_r]	1.778	0.4375				
56	β11	Image Scenario[Random Background with Real Environment]	1.778	0.4375				
57	β12	Image Scenario[Segmented_image]	1.778	0.4375				

Classification Accuracy ~ Intercept + Model Name + Image Scenario + Image Class

Generate the analyse and interpretation report includes Estimates and P Value for each variables

Figure 8. Multiple Linear Regression Workflow and Example of Interpretations.



Parameter estimates	Variable	Estimate	Standard error	95% CI (asymptotic)	|t|	P value	P value sumr
β0	Intercept	0.8642	0.02238	0.8202 to 0.9082	38.62	<0.0001	****
β1	Model Name[EfficentNetB3 Best Weight]	-0.006667	0.01445	-0.03505 to 0.02171	0.4615	0.6446	ns
β2	Model Name[DenseNet121Best Weight]	0.05000	0.01445	0.02162 to 0.07838	3.461	0.0006	***
β3	Model Name[MobileNetV2 Best Weight]	0.04000	0.01445	0.01162 to 0.06838	2.769	0.0058	**
β4	Model Name[ResNet50 Best Weight]	0.03444	0.01445	0.006065 to 0.06282	2.384	0.0175	*
β5	Image Scenario[blur_background]	-0.04250	0.01938	-0.08058 to -0.004424	2.193	0.0288	*
β6	Image Scenario[blur_object]	-0.07000	0.01938	-0.1081 to -0.03192	3.612	0.0003	***
β7	Image Scenario[image_g]	-0.1257	0.01938	-0.1637 to -0.08759	6.484	<0.0001	****
β8	Image Scenario[image_b]	-0.09850	0.01938	-0.1366 to -0.06042	5.082	<0.0001	****
β9	Image Scenario[image_grey]	-0.06517	0.01938	-0.1032 to -0.02709	3.362	0.0008	***
β10	Image Scenario[image_r]	-0.08700	0.01938	-0.1251 to -0.04892	4.489	<0.0001	****
β11	Image Scenario[Random Background with Real Environment]	-0.7078	0.01938	-0.7459 to -0.6698	36.52	<0.0001	****
β12	Image Scenario[Segmented_image]	-0.3012	0.01938	-0.3392 to -0.2631	15.54	<0.0001	****
β13	Image Class[1]	0.1340	0.02238	0.09003 to 0.1780	5.988	<0.0001	****
β14	Image Class[2]	-0.04867	0.02238	-0.09263 to -0.004701	2.175	0.0301	*
β15	Image Class[3]	0.04000	0.02238	-0.003966 to 0.08397	1.787	0.0745	ns
β16	Image Class[4]	0.1004	0.02238	0.05648 to 0.1444	4.488	<0.0001	****
β17	Image Class[5]	0.1333	0.02238	0.08937 to 0.1773	5.958	<0.0001	****
β18	Image Class[6]	0.07667	0.02238	0.03270 to 0.1206	3.426	0.0007	***
β19	Image Class[7]	0.01044	0.02238	-0.03352 to 0.05441	0.4667	0.6409	ns
β20	Image Class[8]	0.09067	0.02238	0.04670 to 0.1346	4.051	<0.0001	****
β21	Image Class[9]	0.09933	0.02238	0.05537 to 0.1433	4.439	<0.0001	****
β22	Image Class[10]	0.1651	0.02238	0.1211 to 0.2091	7.378	<0.0001	****
β23	Image Class[11]	-0.02244	0.02238	-0.06641 to 0.02152	1.003	0.3164	ns

Figure 9. Examples of Multiple Linear Regression Interpretations: (1) $\beta 0$ (Intercept) estimate equal to 0.8642 means that the base classification accuracy when all predictors are at their reference levels is 86.42%. (2) $\beta 2$ (Model Name [DenseNet121 Best Weight]) estimate equal to 0.05000, P value 0.0006 means that the model [DenseNet121 Best Weight] increases the classification accuracy by 5.000% when compared to the reference level the model [EfficentNetB0 Best Weight]. This effect is also statically significant (P value <0.05). (3) $\beta 11$ (Image Scenario [Random Background with Real Environment]) estimate equal to -0.7078; P value <0.0001 means that this image scenario decreases the classification accuracy by 70.78% when compared to the reference level the Image Scenario [Original] with a significant confidence.

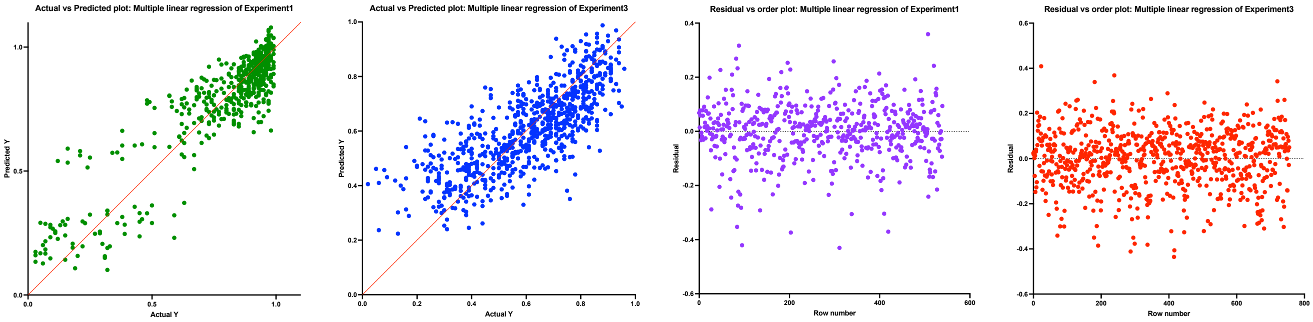
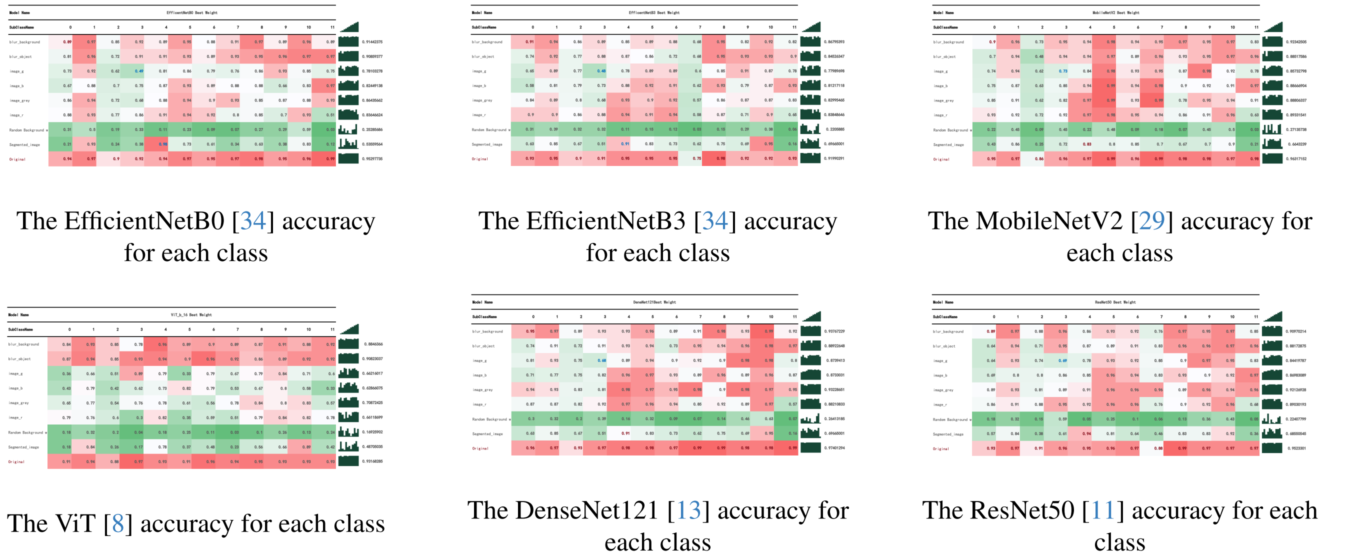


Figure 10. Multiple linear Regression Residual Plot and QQ Plot of Ex1 (SOTA Model Classfication) and Ex3 (SOTA Segmentation) on Hypothesis Verification Process. These shows how two distributions (accuray point)' quantiles line up, with our theoretical distribution (e.g., the normal distribution) as the x variable (Scenarios, Image Object, Model Name) and regression model residuals as the y variable. If the points lie on or close to a 45-degree line, it means that the data follow the reference distribution closely, boosting confidence in the regression results.



The EfficientNetB0 [34] accuracy for each class

The EfficientNetB3 [34] accuracy for each class

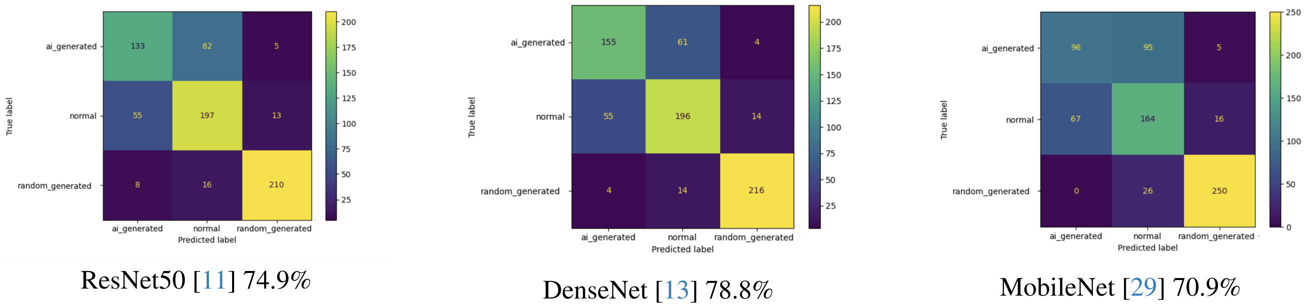
The MobileNetV2 [29] accuracy for each class

The ViT [8] accuracy for each class

The DenseNet121 [13] accuracy for each class

The ResNet50 [11] accuracy for each class

Figure 11. The SOTA models accuracy density map for each class on Experiment 1. Testing images with different background indeed is a challenging scenario for vision models.

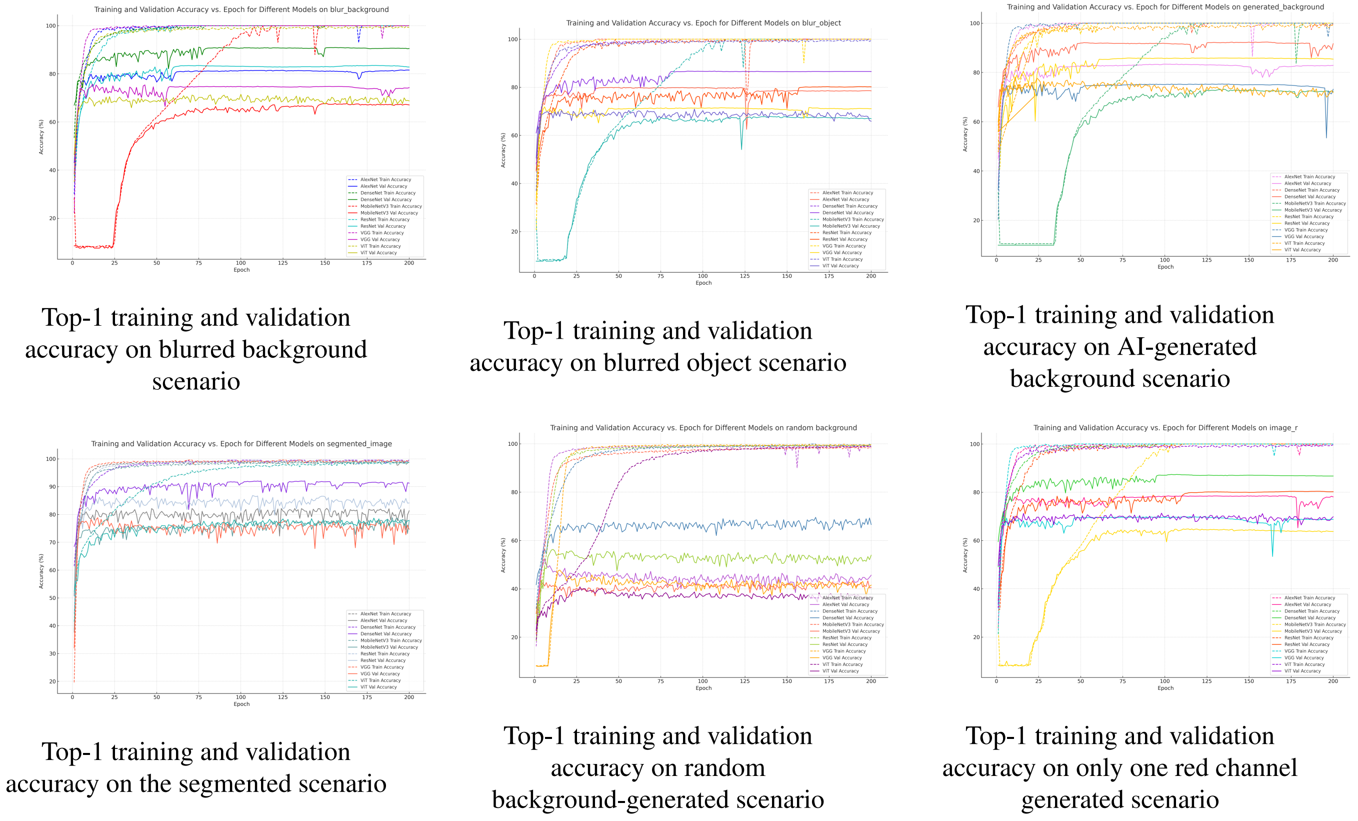


ResNet50 [11] 74.9%

DenseNet [13] 78.8%

MobileNet [29] 70.9%

Figure 12. Model accuracy for classifying normal/AI-generated/random-generated images.



Top-1 training and validation accuracy on blurred background scenario

Top-1 training and validation accuracy on blurred object scenario

Top-1 training and validation accuracy on AI-generated background scenario

Top-1 training and validation accuracy on the segmented scenario

Top-1 training and validation accuracy on random background-generated scenario

Top-1 training and validation accuracy on only one red channel generated scenario

Figure 13. Top-1 training and validation accuracy for SOTA models on Experiment 2: within the same Scenario.

(a) **Prompt**: Generate high-definition pictures like those in the National Geographic magazine, keep the background unchanged.

(b) **Prompt**: Generate a realistic blue sky, and clouds background and please do not change the foreground airship object.

(c) **Prompt**: Generate high resolution images in sea water

(d) **Prompt**: Generate a picture with a foreground and the green grass in the background, similar to the official HD picture released by the state.

(e) **Prompt**: Generate high-resolution pictures like fox in the lawn, National Geographic, keep the background and foreground more simple and real.

(f) **Prompt**: Generate a simple image more realistic in the style of ocean magazine.

(g) **Prompt**: Generate an image with the car in the background and similar to the HD image published by the state.

(h) **Prompt**: Generate high-resolution pictures like those in National Financial Magazine, and keep the background and foreground consistent and the environment more real!

(i) **Prompt**: Please generate high-resolution pictures like those in the National Music Magazine and keep the background unchanged.

(j) **Prompt**: Generate high-resolution pictures in the style of those in National Food Magazine, and keep the background and foreground consistent and the environment more real!

(k) **Prompt**: Generate high-definition pictures like those in the National Geographic magazine, keep the background unchanged.

(l) **Prompt**: Generate high-resolution pictures, such as National Marine Magazine's oceans and whales, to keep the background real.

Figure 14. AI generated images with prompts within XIMAGENET-12 Dataset.