Supplementary Tables

Study design (Part 1)	**Completed,**	**page**
The clinical problem in which the model will be employed is clearly detailed in the paper.	yes	3
The research question is clearly stated.	yes	4
The characteristics of the cohorts (training and test sets) are detailed in the text.	yes	5
The cohorts (training and test sets) are shown to be representative of real-world clinical settings.	yes	5
The state-of-the-art solution used as a baseline for comparison has been identified and detailed.	yes	6
Data and optimization (Parts 2, 3)		
The origin of the data is described and the original format is detailed in the paper.	yes	5
Transformations of the data before it is applied to the proposed model are described.	yes	5
The independence between training and test sets has been proven in the paper.	yes	5
Details on the models that were evaluated and the code developed to select the best model are provided.	yes	36
Is the input data type structured or unstructured?: **Unstructured images**		
Model performance (Part 4)		
The primary metric selected to evaluate algorithm performance (e.g., AUC, F-score, etc.), including the justification for selection, has been clearly stated.	yes	6-7
The primary metric selected to evaluate the clinical utility of the model (e.g., PPV, NNT, etc.), including the justification for selection, has been clearly stated.	-	-
The performance comparison between baseline and proposed model is presented with the appropriate statistical significance.	yes	24
Model examination (Part 5)		
Examination technique 1a: **Highly scoring tiles (qualitative and quantitative analysis)**	yes	14
Examination technique 2a: **Principal component analysis (PCA)**	yes	16
A discussion of the relevance of the examination results with respect to model/algorithm performance is presented.	yes	11
A discussion of the feasibility and significance of model interpretability at the case level if examination methods are uninterpretable is presented.	yes	11
A discussion of the reliability and robustness of the model as the underlying data distribution shifts is included.	yes	11
Reproducibility (Part 6)		
Tier 1: complete sharing of the code	yes	7

Suppl. Table 1: Minimum information about clinical artificial intelligence modeling (MI-CLAIM) checklist

Task	Run	Normal model		Adversarially trained model		
		ResNet	ViT	ResNet	ResNet with DBN	ViT
RCC subtyping *raw data for Figure 1C*	AUROC of run #1	0.967 [0.953 - 0.978]	0.965 [0.952 - 0.976]	0.953 [0.939 - 0.967]	0.926 [0.907 - 0.945]	0.938 [0.916 - 0.955]
	AUROC of run #2	0.975 [0.964 - 0.984]	0.962 [0.949 - 0.973]	0.958 [0.943 - 0.971]	0.962 [0.950 - 0.972	0.942 [0.922 - 0.956]
	AUROC of run #3	0.948 [0.931 - 0.965]	0.941 [0.923- 0.958]	0.956 [0.941 - 0.970]	0.901 [0.878 - 0.920]	0.933 [0.913 - 0.949]
	AUROC of run #4	0.955 [0.937 - 0.970]	0.953 [0.937 - 0.967]	0.947 [0.931 - 0.939]	0.968 [0.955 - 0.980]	0.930 [0.910 - 0.947]
	AUROC of run #5	0.957 [0.943 - 0.971]	0.971 [0.959 - 0.981]	0.954 [0.939 - 0.968]	0.971 [0.959 - 0.982]	0.949 [0.930 - 0.963]
	Mean AUROC +/- SD	0.960 [± 0.009]	0.958 [± 0.010]	0.954 [± 0.004]	0.946 [± 0.028]	0.938 [± 0.007]
	Median AUROC +/- IQR	0.957 [± 0.012]	0.962 [± 0.012]	0.954 [± 0.003]	0.962 [± 0.042]	0.938 [± 0.009]
Gastric cancer subtyping *raw data for Figure 1E*	AUROC of run #1	0.785 [0.726 - 0.844]	0.772 [0.708 - 0.830]	0.767[0.701 - 0.828	0.767[0.692 - 0.837]	0.747 [0.680 - 0.811]
	AUROC of run #2	0.796 [0.733 - 0.858]	0.787 [0.723 - 0.843]	0.765 [0.703 - 0.827]	0.553 [0.477 - 0.636]	0.720 [0.651 - 0.786]
	AUROC of run #3	0.758 [0.690 - 0.823]	0.745 [0.684 - 0.812]	0.756 [0.690 - 0.821]	0.773 [0.708 - 0.831]	0.734 [0.667 - 0.801]
	AUROC of run #4	0.795 [0.731 - 0.858]	0.759 [0.696 - 0.821]	0.748 [0.690 - 0.808]	0.697 [0.618 - 0.769]	0.740 [0.676 - 0.807]
	AUROC of run #5	0.774 [0.708 - 0.834]	0.779 [0.719 - 0.839]	0.735 [0.668 - 0.802]	0.800 [0.743 - 0.854]	0.745 [0.678 - 0.808]
	Mean AUROC +/- SD	0.782 [± 0.014]	0.768 [± 0.015]	0.754 [± 0.012]	0.718 [± 0.089]	0.737 [± 0.010]
	Median AUROC +/- IQR	0.785 [± 0.021]	0.772 [± 0.02]	0.756 [± 0.017]	0.767 [0.076]	0.740 [± 0.011]

Suppl. Table 2: Baseline performance for ResNet and ViT on both classification tasks, no attack at inference. SD = standard deviation, IQR = interquartile range. Adversarially robust training was performed using PGD attack with ε = 1.5e-3.

		Is the noise detectable for a human observer?	
		undetectable	detectable
PGD attack on ResNet *Raw data for Suppl. Figure 2A*	Sum	54	96
	0.0 < ε < 0.1	30	0
	0.1 < ε < 0.2	23	7
	0.2 < ε < 0.3	1	29
	0.3 < ε < 0.4	0	30
	0.4 < ε < 0.5	0	30
PGD attack on ViT *Raw data for Suppl. Figure 2B*	Sum	38	112
	0.0 < ε < 0.1	30	0
	0.1 < ε < 0.2	8	22
	0.2 < ε < 0.3	0	30
	0.3 < ε < 0.4	0	30
	0.4 < ε < 0.5	0	30

Suppl. Table 3: Results of the blinded observer study. The number of images classified in the blinded observer as detectable and undetectable noise for both ResNet and ViT models.

Experimental setup	Experimental run	Attack strength at inference			
		None, ε = 0.0e-3	Low, ε = 2.5e-3	Medium, ε = 7.5e-3	High, ε = 1.50e-3
Model: ResNet Task: RCC subtyping Train: normal Inference: PGD attack	AUROC of run #1	0.967 [0.953 - 0.978]	0.883 [0.853 - 0.908]	0.494 [0.450 - 0.539]	0.058 [0.043 - 0.075]
	AUROC of run #2	0.975 [0.964 - 0.984]	0.940 [0.921 - 0.956]	0.769 [0.729 -0.804]	0.331 [0.291 - 0.375]
	AUROC of run #3	0.948 [0.931 - 0.965]	0.930 [0.908 - 0.949]	0.869 [0.838 - 0.897]	0.734 [0.693 - 0.772]
	AUROC of run #4	0.955 [0.937 - 0.970]	0.921 [0.896 - 0.942]	0.804 [0.767 - 0.837]	0.499 [0.455 - 0.543]
	AUROC of run #5	0.957 [0.943 - 0.971]	0.923 [0.901 - 0.943]	0.809 [0.772 - 0.842]	0.525 [0.476 - 0.570]
	Mean AUROC +/- SD	0.960 [± 0.009]	0.919 [±0.019]	0.749 [± 0.131]	0.429 [± 0.226]
	Median AUROC +/- IQR	0.957 [± 0.012]	0.923 [± 0.009]	0.804 [± 0.04]	0.499 [± 0.194]
Model: ViT Task: RCC subtyping Train: normal Inference: PGD attack	AUROC of run #1	0.965 [0.952 - 0.976]	0.954 [0.939 - 0.968]	0.925 [0.903 - 0.944]	0.861 [0.829 - 0.889]
	AUROC of run #2	0.962 [0.949 - 0.973]	0.947 [0.930 - 0.962]	0.911 [0.888 - 0.932]	0.831 [0.798 - 0.860]
	AUROC of run #3	0.941 [0.923- 0.958]	0.929 [0.911 - 0.948]	0.900 [0.876 - 0.923]	0.844 [0.813 - 0.874]
	AUROC of run #4	0.953 [0.937 - 0.967]	0.933 [0.914 - 0.950]	0.830 [0.855 - 0.906]	0.768 [0.729 - 0.801]
	AUROC of run #5	0.971 [0.959 - 0.981]	0.958 [0.943 - 0.970]	0.922 [0.901 - 0.940]	0.833 [0.801 - 0.863]
	Mean AUROC +/- SD	0.958 [± 0.01]	0.944 [± 0.011]	0.908 [± 0.015]	0.827 [± 0.032]
	Median AUROC +/- IQR	0.962 [± 0.012]	0.947 [± 0.021]	0.911 [± 0.022]	0.833 [± 0.013]
	p versus ResNet	p = 0.98, t = 0.28	p = 0.06, t =-2.21	p = 0.04, t = -2.41	p = 0.01, t = -3.49

Suppl. Table 4: Performance of ResNet and ViT on the RCC subtyping task, attacked with PGD at inference. We used a two-sided t-test without adjustments for the performance comparison between two models

Experimental setup	Experimental run	Attack strength at inference			
		None, ε = 0.0e-3	Low, ε = 0.25e-3	Medium, ε = 0.75e-3	High, ε = 1.50e-3
Model: ResNet Task: Gastric cancer subtyping Train: normal Inference: PGD attack	AUROC of run #1	0.785 [0.726 - 0.844]	0.326 [0.259 - 0.398]	0.006 [0.002 - 0.014]	0.000 [0.000 - 0.000]
	AUROC of run #2	0.796 [0.733 - 0.858]	0.363 [0.292 - 0.435]	0.007 [0.002 - 0.014]	0.000 [0.000 - 0.000]
	AUROC of run #3	0.758 [0.690 - 0.823]	0.519 [0.435 - 0.597]	0.112 [0.074 - 0.153]	0.001 [0.000 - 0.002]
	AUROC of run #4	0.795 [0.731 - 0.858]	0.401 [0.329 - 0.477]	0.017 [0.006 - 0.033]	0.000 [0.000 - 0.000]
	AUROC of run #5	0.774 [0.708 - 0.834]	0.293 [0.225 - 0.364]	0.003 [0.000 - 0.008]	0.000 [0.000 - 0.000]
	Mean AUROC +/- SD	0.782 [± 0.014]	0.380 [± 0.078]	0.029 [± 0.042]	0.000 [± 0.000]
	Median AUROC +/- IQR	0.785 [± 0.021]	0.363 [± 0.075]	0.007 [± 0.011]	0.000 [± 0.000]
Model: ViT Task: Gastric cancer subtyping Train: normal Inference: PGD attack	AUROC of run #1	0.772 [0.708 - 0.830]	0.676 [0.605 - 0.747]	0.437 [0.361 - 0.513]	0.175 [0.123 - 0.237]
	AUROC of run #2	0.787 [0.723 - 0.843]	0.726 [0.657 - 0.791]	0.589 [0.510 - 0.664]	0.382 [0.305 - 0.454]
	AUROC of run #3	0.745 [0.684 - 0.812]	0.698 [0.633 - 0.767]	0.600 [0.527 - 0.673]	0.446 [0.375 - 0.524]
	AUROC of run #4	0.759 [0.696 - 0.821]	0.581 [0.506 - 0.654]	0.255 [0.190 - 0.322]	0.028 [0.011 - 0.05]
	AUROC of run #5	0.779 [0.719 - 0.839]	0.729 [0.666 - 0.794]	0.631 [0.560 - 0.704]	0.459 [0.386 - 0.545]
	Mean AUROC +/- SD	0.768 [± 0.015]	0.682 [± 0.054]	0.502 [± 0.141]	0.298 [± 0.169]
	Median AUROC +/- IQR	0.772 [± 0.02]	0.698 [± 0.050]	0.589 [± 0.163]	0.382 [± 0.271]
	P-value ResNet vs. ViT	p = 0.24, t = 1.28	p = 0.00, t = - 6.35	p = 0.00, t = - 6.45	p = 0.01, t = - 3.52

Suppl. Table 5: Performance of ResNet and ViT on the gastric cancer subtyping task, attacked with PGD at inference. We used a two-sided t-test without adjustments for the performance comparison between two models.

Experimental setup	Experimental run	Attack strength at inference			
		None, ε = 0.0e-3	Low, ε = 0.25e-3	Medium, ε = 0.75e-3	High, ε = 1.50e-3
Model: ResNet Task: RCC subtyping Train: adversarially robust training Inference: PGD attack	AUROC of run #1	0.953 [0.939 - 0.967]	0.950 [0.935 - 0.965]	0.943 [0.926 - 0.959]	0.930 [0.912 - 0.948]
	AUROC of run #2	0.958 [0.943 - 0.971]	0.955 [0.940 - 0.969]	0.949 [0.933 - 0.964]	0.940 [0.921 - 0.957]
	AUROC of run #3	0.956 [0.941 - 0.970]	0.953 [0.937 - 0.968]	0.946 [0.928 - 0.961]	0.934 [0.915 - 0.951]
	AUROC of run #4	0.947 [0.931 - 0.939]	0.944 [0.928 - 0.959]	0.938 [0.921 - 0.954]	0.929 [0.910 - 0.946]
	AUROC of run #5	0.954 [0.939 - 0.968]	0.951 [0.935 - 0.965]	0.943 [0.926 - 0.959]	0.928 [0.909 - 0.947]
	Mean AUROC +/- SD	0.954 [± 0.004]	0.951 [± 0.004]	0.944 [± 0.004]	0.932 [± 0.004]
	Median AUROC +/- IQR	0.954 [± 0.003]	0.951 [± 0.003]	0.943 [± 0.003]	0.930 [± 0.005]
Model: ViT Task: RCC subtyping Train: adversarially robust training Inference: PGD attack	AUROC of run #1	0.938 [0.916 - 0.955]	0.934 [0.912 - 0.952]	0.926 [0.903 - 0.945]	0.913 [0.889 - 0.933]
	AUROC of run #2	0.942 [0.922 - 0.956]	0.936 [0.916 - 0.952]	0.929 [0.907 - 0.946	0.915 [0.892- 0.935
	AUROC of run #3	0.933 [0.913 - 0.949]	0.928[0.907 - 0.945]	0.918 [0.896 - 0.937]	0.903 [0.878 - 0.923]
	AUROC of run #4	0.930 [0.910 - 0.947]	0.921 [0.900 - 0.939]	0.904 [0.880 - 0.925]	0.876 [0.848 - 0.900]
	AUROC of run #5	0.949 [0.930 - 0.963]	0.943 [0.924 - 0.959]	0.935 [0.914 - 0.952]	0.921 [0.898 - 0.941]
	Mean AUROC +/- SD	0.938 [± 0.007]	0.932 [± 0.007]	0.922 [± 0.01]	0.906 [± 0.016]
	Median AUROC +/- IQR	0.938 [± 0.009]	0.934 [± 0.008]	0.926 [± 0.011]	0.913 [± 0.012]
	p versus ResNet	p = 0.00-ä , t = 3.96	p = 0.00, t = 4.37	p = 0.01, t = 3.78	p = 0.01, t = 3.23
Model: ResNet Task: RCC subtyping Train: adversarially robust DBN Inference: PGD attack	AUROC of run #1	0.926 [0.907 - 0.945]	0.927 [0.908 - 0.945]	0.907 [0.886 - 0.929]	0.873 [0.847 - 0.899]
	AUROC of run #2	0.962 [0.950 - 0.972	0.960 [0.947 - 0.972]	0.951 [0.936 - 0.965]	0.937 [0.919 - 0.953]
	AUROC of run #3	0.901 [0.878 - 0.920]	0.859 [0.832 - 0.883	0.830 [0.800 - 0.857]	0.777 [0.742 - 0.808]
	AUROC of run #4	0.968 [0.955 - 0.980]	0.966 [0.952 - 0.978]	0.953 [0.936 - 0.968]	0.929 [0.907 - 0.949]
	AUROC of run #5	0.971 [0.959 - 0.982]	0.960 [0.944 - 0.973]	0.948[0.931 - 0.964]	0.927 [0.906 - 0. 945]
	Mean AUROC +/- SD	0.946 [± 0.028]	0.934 [± 0.04]	0.918 [± 0.047	0.889 [± 0.06]
	Median AUROC +/- IQR	0.962 [± 0.042]	0.960 [± 0.033]	0.948 [± 0.044]	0.927 [± 0.056]
	p versus ResNet	p = 0.58, t = 0.58	p = 0.44, t = 0.80	p = 0.30, t = 1.1	p = 0.19, t = 1.44

Suppl. Table 6: Performance of adversarially robustly trained ResNet and ViT on the RCC subtyping task, attacked with PGD at inference. We used a two-sided t-test without adjustments for the performance comparison between two models.

		Attack strength at inference

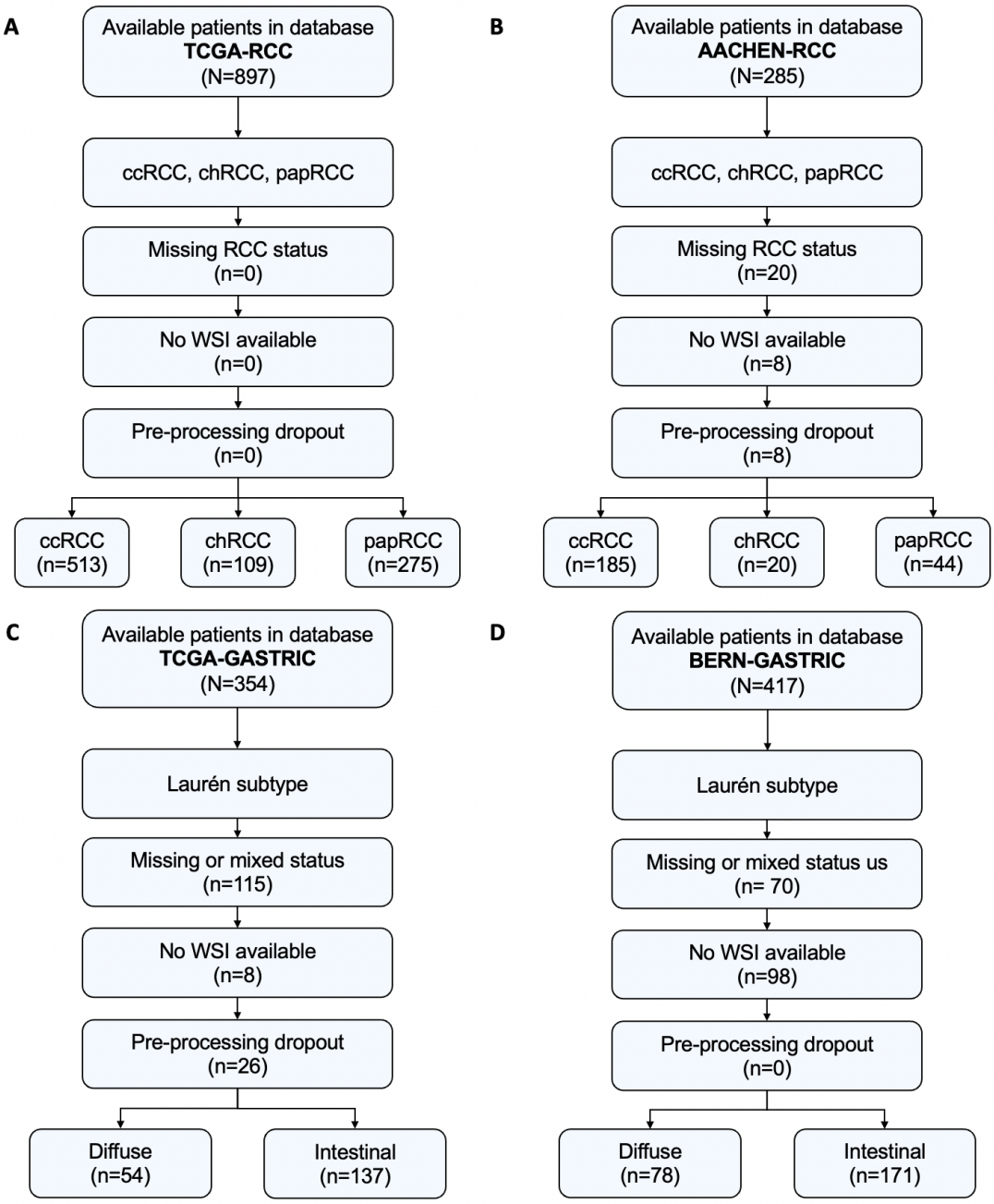
Experimental setup	Experimental run	None, ε = 0.0e-3	Low, ε = 0.25e-3	Medium, ε = 0.75e-3	High, ε = 1.50e-3
Model: ResNet Task: Gastric cancer subtyping Train: adversarially robust training Inference: PGD attack	AUROC of run #1	0.767[0.701 - 0.828	0.739 [0.671 - 0.803]	0.680 [0.608 - 0.751]	0.588 [0.510 - 0.668]
	AUROC of run #2	0.765 [0.703 - 0.827]	0.741 [0.676 - 0.807]	0.683 [0.613 - 0.755]	0.581 [0.504 - 0.657]
	AUROC of run #3	0.756 [0.690 - 0.821]	0.738 [0.668 - 0.803]	0.701 [0.631 - 0.766]	0.638 [0.563 - 0.708]
	AUROC of run #4	0.748 [0.690 - 0.808]	0.722 [0.662 - 0.785]	0.663 [0.599- 0.733]	0.566 [0.498 - 0.641]
	AUROC of run #5	0.735 [0.668 - 0.802]	0.713 [0.644 - 0.783]	0.667 [0.596 - 0.742]	0.604 [0.529 - 0.685]
	Mean AUROC +/- SD	0.754 [± 0.012]	0.731 [± 0.011]	0.679 [± 0.013]	0.595 [± 0.025]
	Median AUROC +/- IQR	0.756 [± 0.017]	0.738 [± 0.017]	0.680 [± 0.016]	0.588 [± 0.023]
Model: ViT Task: Gastric cancer subtyping Train: adversarially robust training Inference: PGD attack	AUROC of run #1	0.747 [0.680 - 0.811]	0.731 [0.662 - 0.798]	0.698 [0.625 - 0.768]	0.649 [0.574 - 0.722]
	AUROC of run #2	0.720 [0.651 - 0.786]	0.705 [0.634 - 0.771]	0.681 [0.608 - 0.748]	0.642 [0.567 - 0.715]
	AUROC of run #3	0.734 [0.667 - 0.801]	0.721 [0.652 - 0.787]	0.695 [0.621 - 0.763]	0.652 [0.577 - 0.725]
	AUROC of run #4	0.740 [0.676 - 0.807]	0.728 [0.664 - 0.797]	0.705 [0.639 - 0.775]	0.667 [0.598 - 0.739]
	AUROC of run #5	0.745 [0.678 - 0.808]	0.734 [0.667 - 0.799]	0.714 [0.645 - 0.779]	0.674 [0.598 - 0.741]
	Mean AUROC +/- SD	0.737 [± 0.010]	0.724 [± 0 .010]	0.699 [± 0.011]	0.657 [± 0.012]
	Median AUROC +/- IQR	0.740 [± 0.011]	0.728 [± 0.010]	0.698 [± 0.010]	0.652 [± 0.018]
	P-value ResNet vs. ViT	p = 0.06, t = 2.23	p = 0.40, t = 0.90	p = 0.05, t = -2.28	p = 0.00, t = -4.5

Suppl. Table 7: Performance of adversarially robustly trained ResNet and ViT on the gastric cancer subtyping task, attacked with PGD at inference. We used a two-sided t-test without adjustments for the performance comparison between two models.

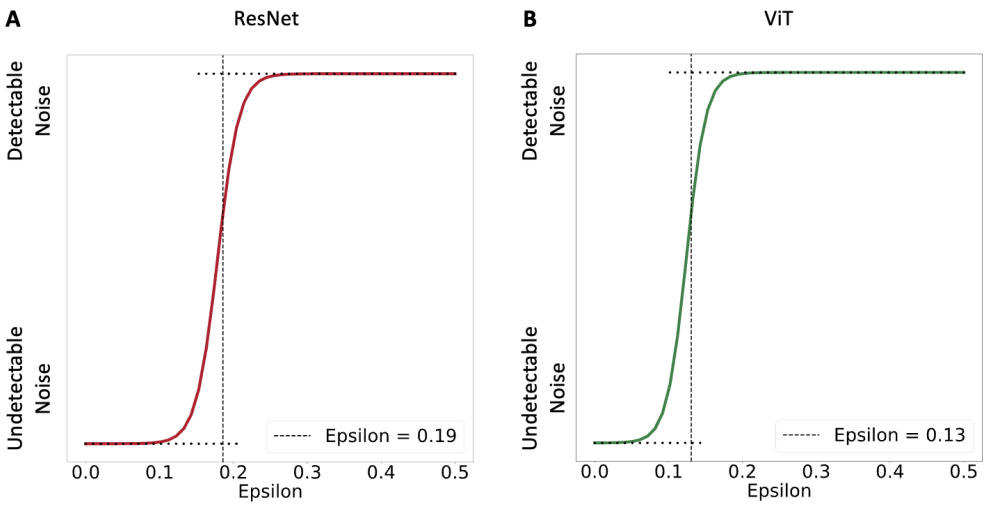
Metric	Task	Class	Original images			Attacked images		
			ResNet	**ViT**	***Ratio ViT / ResNet***	**ResNet**	**ViT**	***Ratio ViT / ResNet***
Spread within the class (distance from points to center) *lower is better*	**RCC**	**ccRCC**	0.158	**0.056**	*0.356*	0.120	**0.051**	*0.425*
		chRCC	**0.057**	0.064	*1.123*	0.123	**0.115**	*0.935*
		papRCC	0.171	**0.043**	*0.251*	0.169	**0.087**	*0.515*
	Gastric cancer	**diffuse**	0.214	**0.206**	*0.963*	**0.174**	0.266	*1.529*
		intestinal	0.216	**0.175**	*0.810*	**0.184**	0.260	*1.413*
Spread between classes (distance from center to center) *higher is better*	**RCC**	**ccRCC to chRCC**	0.478	**0.899**	*1.860*	0.459	**0.842**	*1.834*
		ccRCC to papRCC	0.560	**0.922**	*1.646*	0.457	**0.896**	*1.961*
		chRCC to papRCC	0.524	**0.882**	*1.683*	0.434	**0.751**	*1.730*
	Gastric cancer	**diffuse to intestinal**	0.502	**0.629**	*1.253*	0.295	**0.634**	*2.149*

Suppl. Table 8. **Spread of data points within classes, and distance between classes in the latent space, related to Figure 5A-B**. The average Euclidean distance of the points within each cluster to its center for features extracted from the original images. Also it reports the distance between the center of 3 clusters. For normal images and perturbed images (PGD attack with ε of 0.05). In each pairwise comparison, the better value is printed bold.

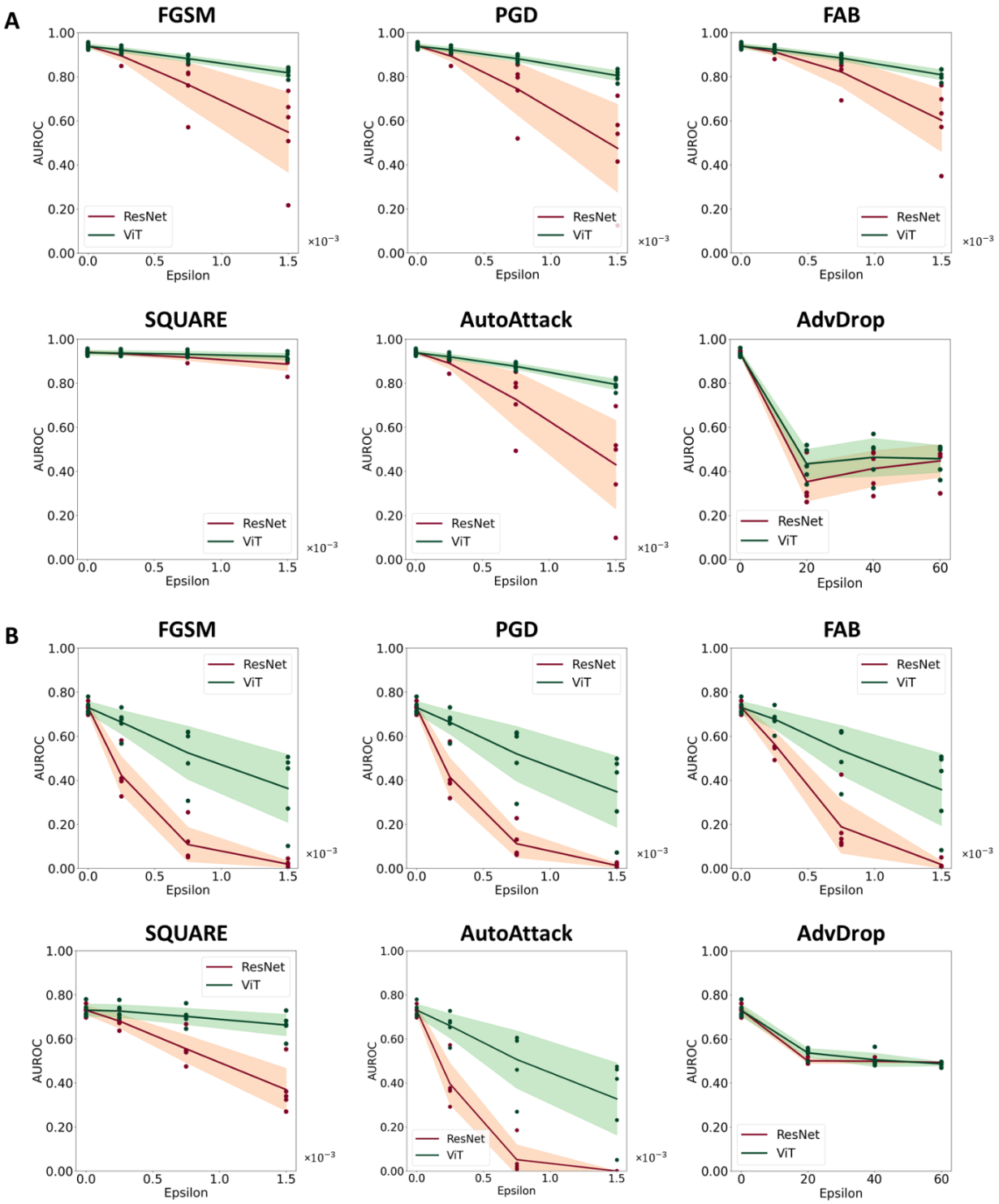
Supplementary Figures



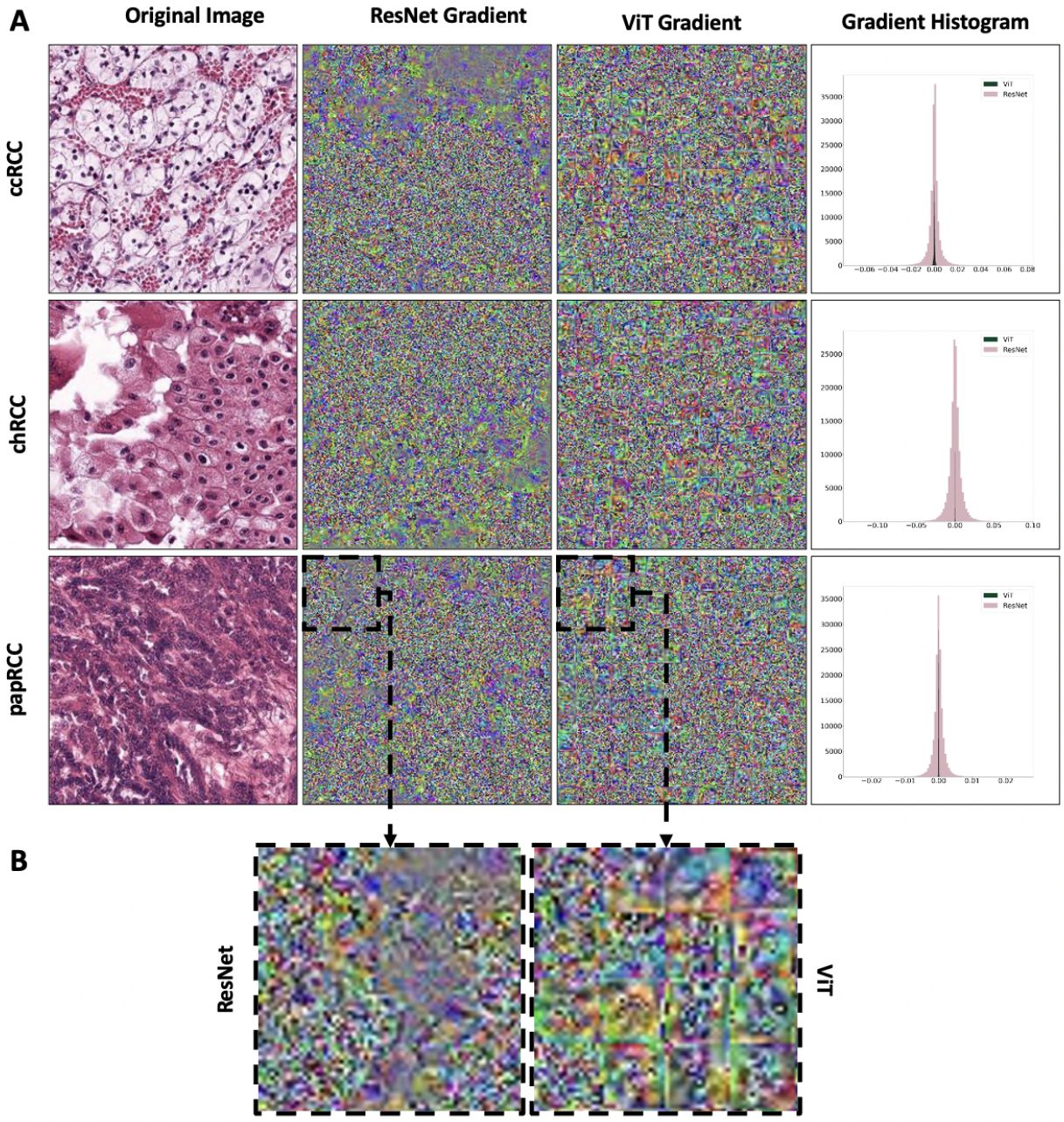
Suppl. Figure 1: CONSORT charts for all cohorts in this study. **(A)** TCGA-RCC, **(B)** AACHEN-RCC, **(C)** TCGA-GASTRIC, **(D)** BERN-GASTRIC. N = total patient number, n = patient number of subset.



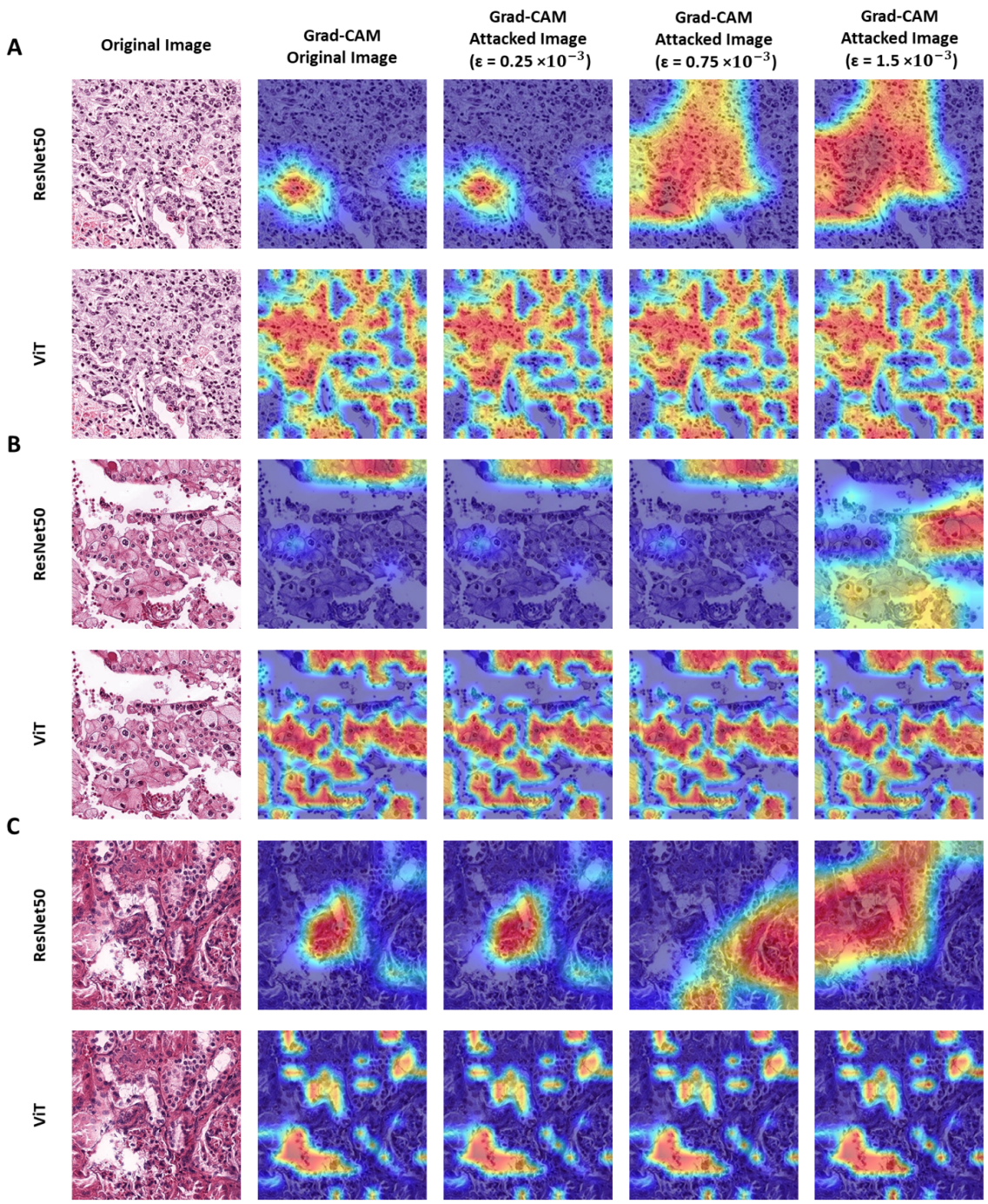
Suppl. Figure 2: Determining the observable threshold for adversarial attacks. (A) In a blinded user study, 150 images with different amounts of noise were assessed. The results were analyzed with a logistic regression. For ResNet, the observable threshold for adversarial noise was ε=0.19. **(B)** For ViT, the observable threshold for adversarial noise was ε=0.13. Raw data for this figure is given in Suppl. Table 3.



Suppl. Figure 3: Micro-averaged AUROC for ResNet and ViT under multiple attack types. (A) For the RCC subtyping task and **(B)** corresponding result for the gastric cancer subtyping task. For computational efficiency, these experiments were run on 10 randomly selected tiles for each whole slide image. We used the Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), Fast Adaptive boundary (FAB), Square attacks, AutoAttack (AA) and AdvDrop for these experiments. This figures show the mean AUROC value of 5 experiments $\pm$ standard deviation.



Suppl. Figure 4: Visualization of PGD gradients for representative images. (A) One representative image for each class in the RCC subtyping task was used to visualize the PGD gradients obtained with a white-box attack on a ResNet and a ViT model. **(B)** The enlarged detail shows that the adversarial noise follows the patch structure of the Transformer.



Suppl. Figure 5: Region Visualization of Grad-CAM representations for both ResNet50 and ViT models. (A) One representative image for each class in the RCC subtyping task was used to visualize the PGD gradients obtained with a white-box attack on a ResNet and a ViT model. **(B)** The enlarged detail shows that the adversarial noise follows the patch structure of the Transformer.

Supplementary Methods

Deep Learning models

We used two types of Deep Learning models: Residual neural networks (ResNets) and Vision Transformers (ViTs).

ResNets were developed in 2015 to overcome the vanishing gradient effect problem when training deep neural networks. In addition to possessing all common elements of typical convolutional neural networks such as convolution, pooling, activation and fully connected layers, this architecture also uses identity connection between residual blocks to add the output from the previous layer to the layer ahead of it. Resnet architectures are currently the state of the art in computational pathology [23]. We used ResNet50 pre-trained on ImageNet as the initial model. The first 50% of the layers were frozen and only the last 50% of them were fine-tuned during the training process. The input for the ResNet50 are tiles with a size of $(224 \times 224 \times 3)$. The output for each image after a softmax layer is the prediction scores belonging to each target class. Additionally, we used ResNet50-v2 (Big Transfer Model, BiT) [69] pretrained on a larger dataset (21K) to have an unbiased comparison to the ViTs which have been trained on a larger dataset. ResNet50-v2 (68 million trainable parameters) while keeping the main architecture of the ResNet50-v1 (23 million trainable parameters), applies Batch Normalization and ReLU activation to the input before the convolution operation.

ViTs [56] were introduced in 2021 for image classification tasks after transformers demonstrated enormous success in natural language processing tasks. ViTs work by splitting the input image into small patches and then by using linear projection, creating patch-embeddings from the flattened patches. In addition to this, it adds positional embedding into the patch-embeddings and feeds this sequence as an input into the standard transformer encoder. As a vision transformer model, we used the 'B_16_imagenet1k' model pre-trained on ImageNet with a minor modification of the input size $(224 \times 224 \times 3)$ from the pytorch_pretrained_vit Python package. As with ResNet, the output of ViT for each tile is the prediction scores for each class.

Hyperparameters

To find the optimum learning rate for the training we used the torch_lr_finder Python package with an initial learning rate of 10^{-5} and weight_decay of 10^{-2} for 100 iterations with an end learning rate of 100. We used Adam optimizer during training with the calculated $\frac{optimum\ learning\ rate}{10}$ and a weight-decay of 10^{-2}. The defined loss function for all experiments was cross-entropy. For all the models (ResNets, ViT, and DBN), we resize the input images to $(224 \times 224 \times 3)$ in order to have a fair comparison between different

architectures. In each training run, we randomly selected 20% of the patients for the validation set to stop the training using early stopping with the minimum training epoch of 10 and the patience of 5.

Types of adversarial attacks

Adversarial attacks can be both white-box attacks in which the attacker has access to the model's parameters, or black box attacks in which the attacker has no access to the parameters and it uses either a different model or no model to perturb the images. Additionally, the attacks can be targeted attacks to force the model to output a specific class prediction for an input image or non-target attacks in which general misclassification is the primary purpose. In this study, we use six untargeted attacks:

1. **Fast Gradient Sign Method (FGSM)** [58–60]. This is a white-box attack which generates an adversarial image by changing the pixel magnitudes in the direction of the gradient. This attack is a single-step attack and is therefore very efficient in terms of computation time.
2. **Projected Gradient Descent (PGD)**. This is a multi-step attack based on FGSM and is considered as one of the most powerful and complete white-box adversaries [61]. PGD generates perturbations to maximize the value of loss function while constraining the changes in a specified value, defined as epsilon (ϵ). Because of its iterative nature, PGD is more time consuming. Models that are robust to PGD are always robust against other forms of gradient-based attacks [70,71].
3. **Fast Adaptive boundary (FAB)** [62]. This white-box attack searches for the minimum perturbation which is required to change the class of an input image. Due to the architecture of FAB, to check the success rate of models, it doesn't require repeating the attack for different epsilon values. This stands in contrast to PGD, as described above. As such, it provides a more complete picture of the robustness of a specific model while additionally requiring more computational time in comparison to other white-box attacks.
4. **Square attack** [63]. Unlike the white-box attacks described above, a square attack is a query-efficient black-box attack which does not rely on local gradient information of the specific model [63]. This iterative black-box adversary places square-shaped updates at random positions on the input image and searches for increases in the loss function in each iteration. While square attacks have the novelty of being significantly different in approach to most other attacks, they usually have a lower success rate and are more computationally expensive than white-box attacks [48,72,73].
5. **Auto Attack (AA)** [74]**.** This attack is the combination of multiple attacks with the aim of forming Parameter-free and user-independent adversarial attack. Standard verison of the AA combines Auto PGD (APGD), FAB and Square attack. The combination of diverse white and black-box attacks increase the attack success rate since if for some classifier, one of the attacks fails to perform, at least one of the others will work well. Additionally, diverse attacks succeed on different

points to change the prediction of the model and the combination of them provides a robust success rate.

6. **AdvDrop** [65]**.** This adversarial attack generates perturbed images with dropping imperceptible details from the input image. This attack converts the input image into the frequency domain and then uses differential quantization to drop a series of specific frequencies and then it inverts the signals into the image. The main difference between this attack and the others is that in AdvDrop, attack gets optimized in the frequency domain and therefore the value of epsilon is the constraint for the quantization table. In this study we used the epsilon values from the original paper.

All the attacks which we used in this study are from https://github.com/Harry24k/adversarial-attacks-pytorch repository. For all attacks except advDrop we used three levels of ε were predefined: low (0.25×10^{-3}), medium (0.75×10^{-3}), and high (1.5×10^{-3}). For AdvDrop we used constraint values of 20, 40, and 60. For all the other parameters, we used predefined default values.

Attack mitigation strategies

We used two attack mitigation strategies: adversarially robust training and dual batch normalization.

To robustly train ResNet and ViT, we attacked the images with a PGD white-box attack (ε = 1.5×10^{-3}, alpha = 0.0025) just before feeding them to the models during the training and subsequently calculated the loss function based on the prediction of models for perturbed images. We evaluated adversarially pre-trained models in concordant attacks (e.g. train with PGD, test with PGD) and discordant attacks (e.g. train with PGD, test with FGSM, FAB, AutoAttack, Square attack, advDrop).

The dual batch adversarial robust model was introduced to the medical field by Han et al. in 2021 [57] . This model is a modified ResNet50 with two batch norm layers, one for the standard input and the other for the adversarially perturbed inputs. We use the same hyperparameters for adversarial attacks in this training. The total loss is the sum of the normal loss function for the standard inputs and the loss function for the perturbed images for the same label.

Hardware

All experiments were run on local computer workstations with Nvidia RTX A6000 and Quadro RTX 8000 graphics processing units (GPUs).