Test-time adaptations and Test-time transformations

- Test-time adaptation methods usually modify weights (θ_{ERM}) to improve confidence of predictions.
  - They often lead to collapsed solutions. Entropy can be minimized by trivial prediction vector.

$$\theta^* = argmin_\theta \; H(\theta(x^*))$$

- In parallel, Test-time augmentations often utilize N augmented copies of inputs and ensemble predictions

$$y^* = \arg\max \frac{1}{N} \sum \theta(x_i^*)$$

- Benefit : Improved classification performance and uncertainty

- Issue with Test-time augmentations : E.g Geometric transforms – are they out of domain for pre-trained model ?
  - What does rotating an image and testing it mean ?

TTA via TTT (Proposed approach)

- Nomenclature : TTA – Test-time adaptation, TTT – Test-time transformations
- Define a set of domain-knowledge infused transformations (e.g. Denoiser on noisy input) - $\{T_i\}_{i=1,2\,..\,N}$
- Adapt models by three-part loss function that enforces confidence on each of the transformed copies and also ensures pairwise consistency on predictions and features is established between transformed copies.
- Let θ_z be partial neural network : typically till last layer of features, excluding final linear layer for class mapping

$$\theta^* = argmin_\theta \frac{1}{N}\sum L_1(\theta(T_i(x^*))$$

$$+\lambda_1 * \frac{1}{N*(N-1)}\sum_{i=i}^{N}\sum_{j=1,i\neq j}^{N} L_2\left(\theta\left(T_i(x^*)\right),\theta\left(T_j(x^*)\right)\right)$$

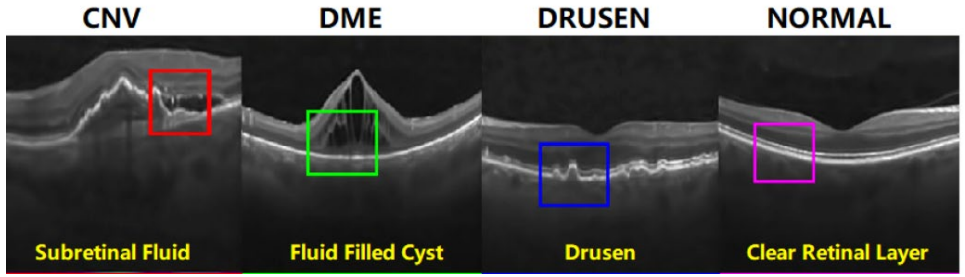
$$+\lambda_2 * \frac{1}{N*(N-1)}\sum_{i=i}^{N}\sum_{j=1,i\neq j}^{N} L_3\left(\theta_z(T_i(x^*)),\theta_z(T_j(x^*))\right)$$

L_1 : Certainty Loss, e.g Entropy
L_2 : Loss on predictions, e.g CE
L_3 : Loss on features, e.g MSE

λ_1, λ_2: scaling factors

Example : OCT Retinal disease classification

- Training : 4 class problem, noiseless OCT images. Testing: Speckle noise corrupted images – 3 levels .



- Key summary from traditional models.
  1. Performance drops on higher noise data for all models, larger the noise level greater the performance drop is.
  2. Performance drops on denoised inputs. Denoised inputs become out-of-distribution for pre-trained model!

Classifier	Noise Free				Speckle (0.7)				Speckle (1.0)		
	P	R	F1	A	P	R	F1	A	P	R	F1
ResNet18	0.96	0.94	0.95	0.97	0.81	0.73	0.75	0.86	0.74	0.64	0.66
MobileNetV2	0.95	0.94	0.94	0.97	0.60	0.58	0.49	0.56	0.54	0.49	0.38
ShuffleNetV2	0.94	0.92	0.93	0.96	0.63	0.47	0.47	0.63	0.61	0.41	0.39
SqueezeNet	0.95	0.93	0.94	0.97	0.57	0.61	0.51	0.59	0.51	0.50	0.40

BM3D denoising

Speckle (0.7)			
P	R	F1	A
0.70	0.58	0.56	0.70
0.49	0.44	0.36	0.48
0.50	0.36	0.30	0.46
0.52	0.53	0.44	0.54

NLM denoising

Speckle (0.7)			
P	R	F1	A
0.66	0.62	0.55	0.66
0.59	0.58	0.49	0.56
0.45	0.36	0.28	0.42
0.60	0.66	0.55	0.63

BILAT denoising

Speckle (0.7)				
P	R	F1	A	P
0.81	0.72	0.74	0.86	0.74
0.59	0.56	0.48	0.56	0.53
0.63	0.44	0.43	0.60	0.62
0.57	0.61	0.51	0.59	0.51

TTA via TTT for OCT retinal disease classification

- Transformations : Denoisers : BM3D, BILAT, NLM
- N = 4, $T_1 = I, T_{2-4}$: Battery of denoisers
- Adapt models on batch-size = 128, steps = 4
- Summary : TTA via TTT improves metrics by 6-9%
  - Even though original model fails on denoised inputs, TTA utilizes them effectively!

Classifier	Noise Free				Speckle (0.7)			
	P	R	F1	A	P	R	F1	A
ResNet18	0.96	0.94	0.95	0.97	0.81	0.73	0.75	0.86
MobileNetV2	0.95	0.94	0.94	0.97	0.60	0.58	0.49	0.56
ShuffleNetV2	0.94	0.92	0.93	0.96	0.63	0.47	0.47	0.63

With TTA via TTT (proposed approach)

Speckle (0.7)			
P	R	F1	A
0.89	0.80	0.84	0.92
0.77	0.80	0.78	0.86
0.65	0.68	0.64	0.75

Next steps

1. Demonstrate better uncertainty measures

$$UN_{ERM} = std\left(\theta\left(T_i(x^*)\right)\right) \; vs \; UN_{TTA} = std\left(\theta^*\left(T_i(x^*)\right)\right)$$

2. Utilize domain specific transforms and demonstrate on CT segmentation.
    - E.g Recon kernel change, window level change

3. Demonstrate visual explainability with gradCAM or other approaches