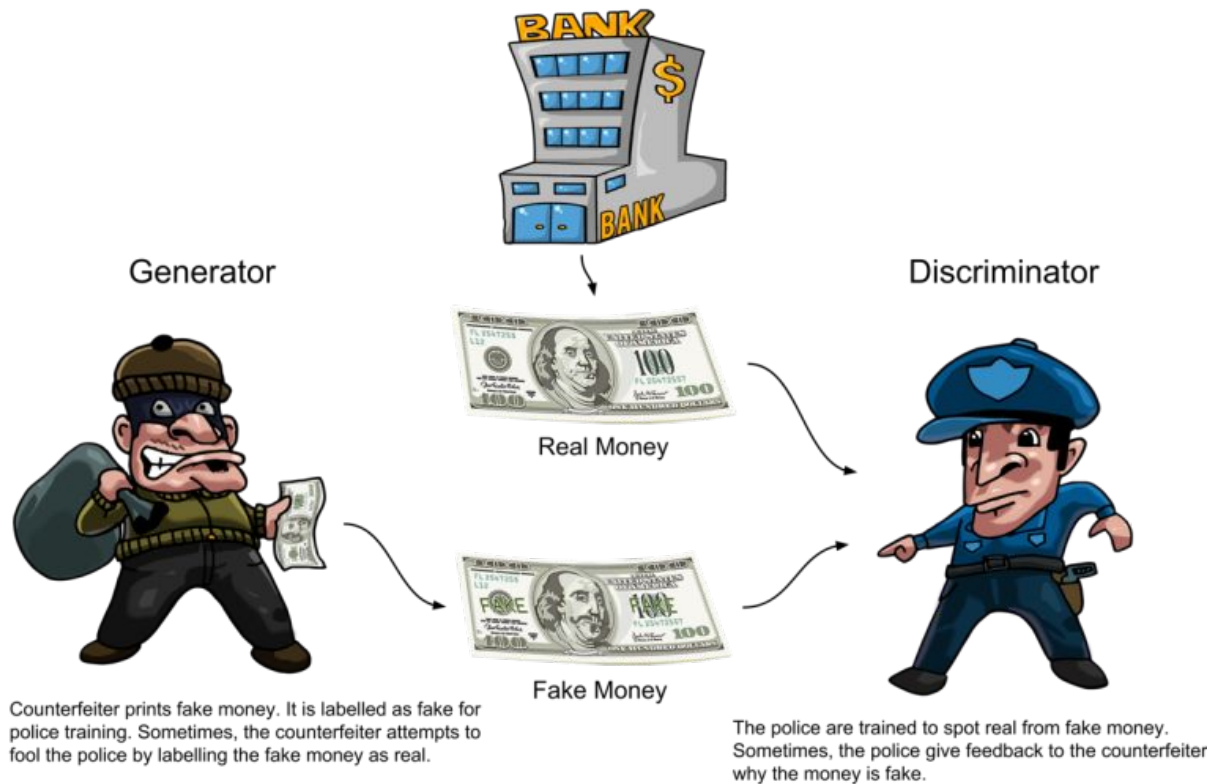


Bài toán Signature Verification sử dụng Discriminator của GAN

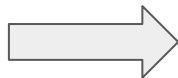
Đặng Tấn Lộc

GAN



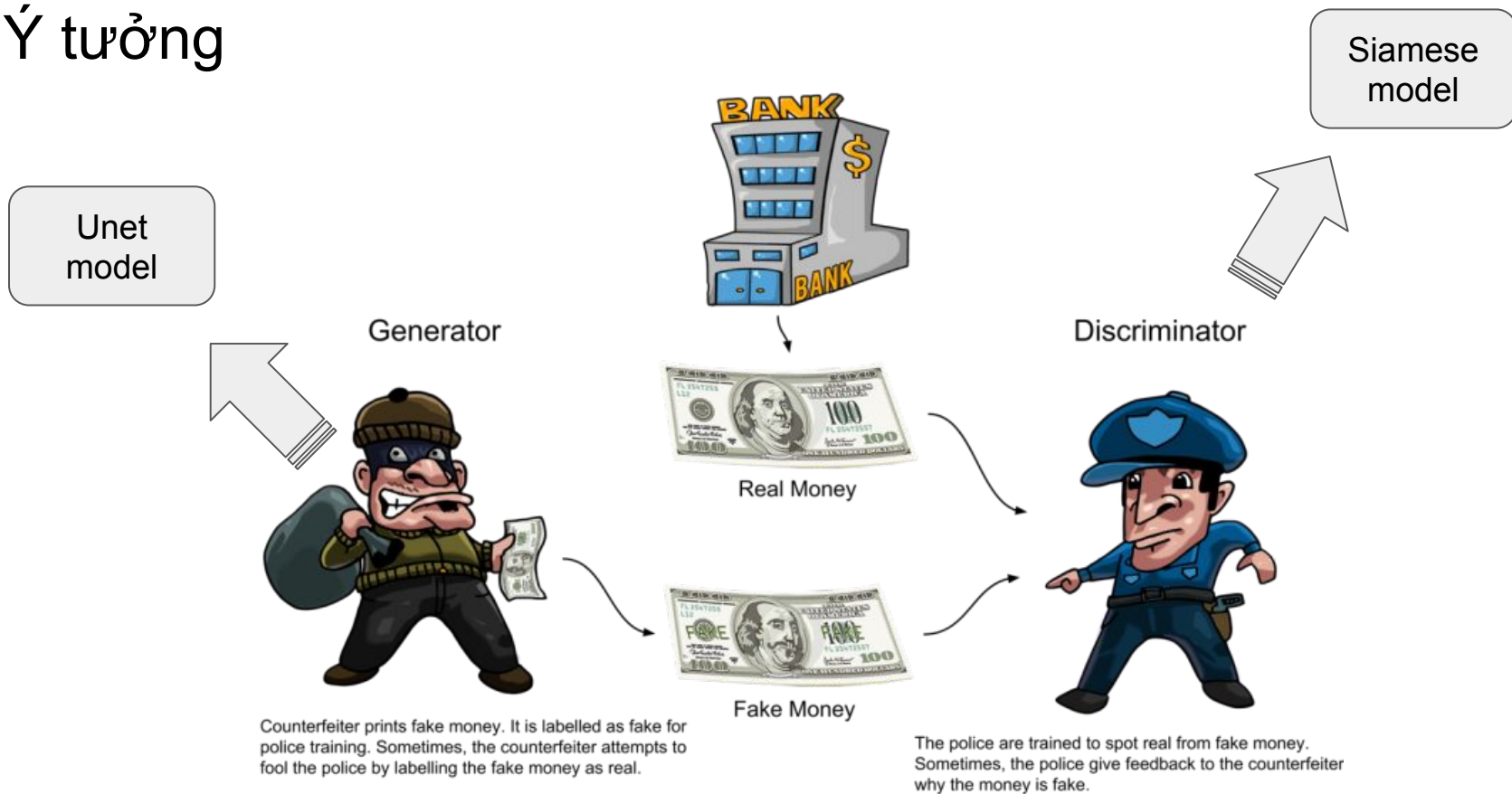
Ý tưởng

Discriminator

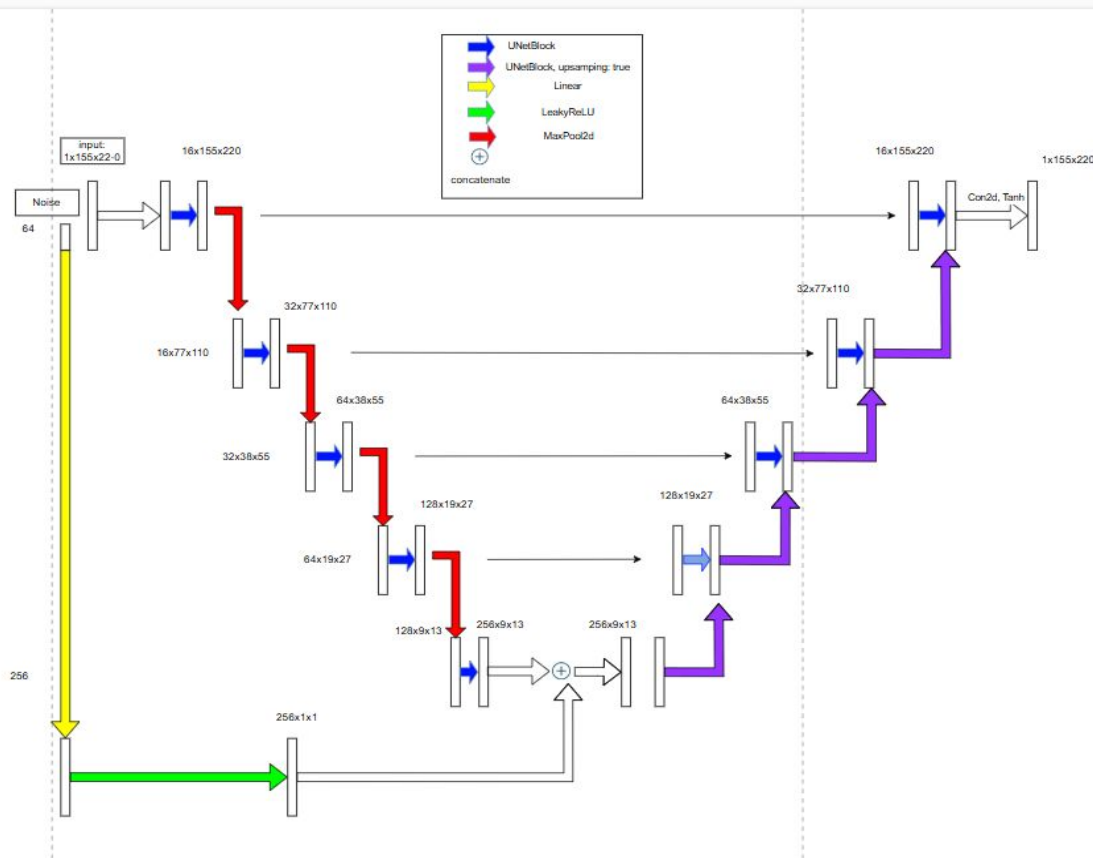


Áp dụng cho bài
toán phân loại

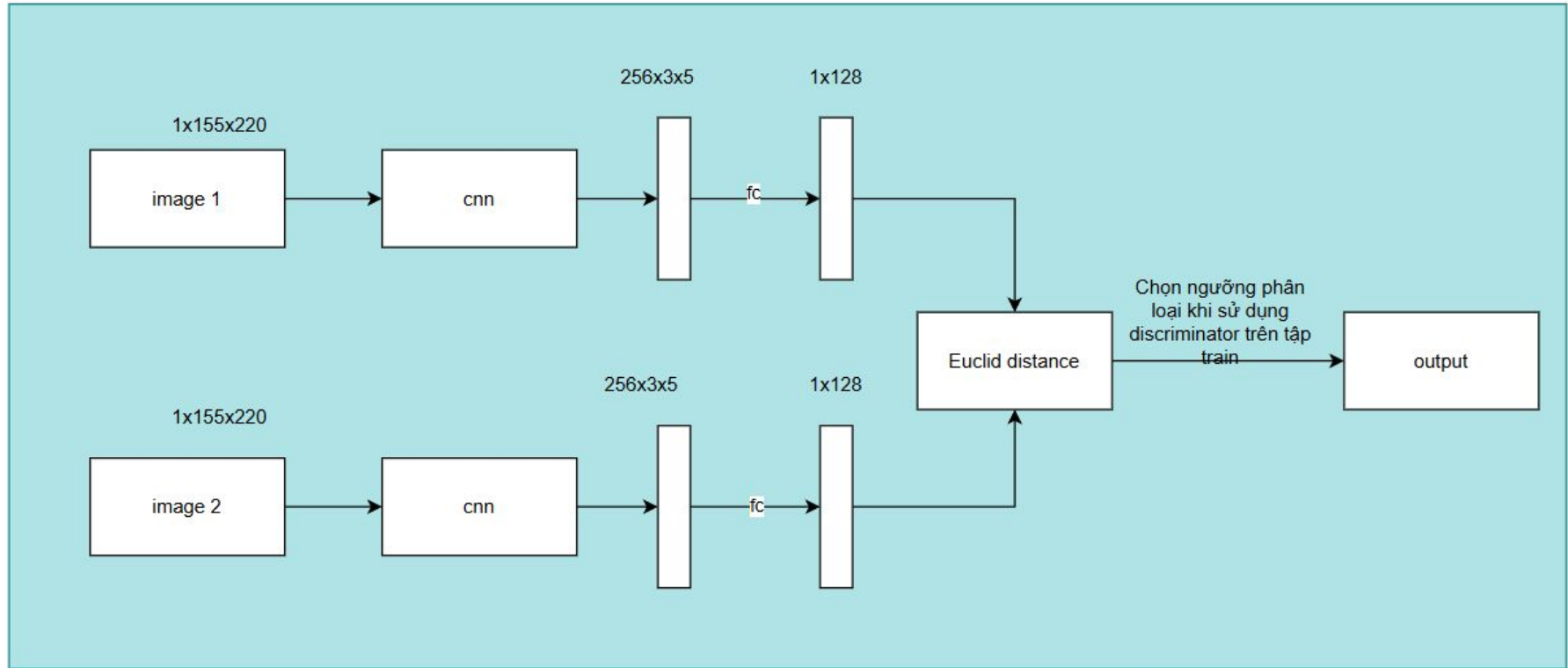
Ý tưởng



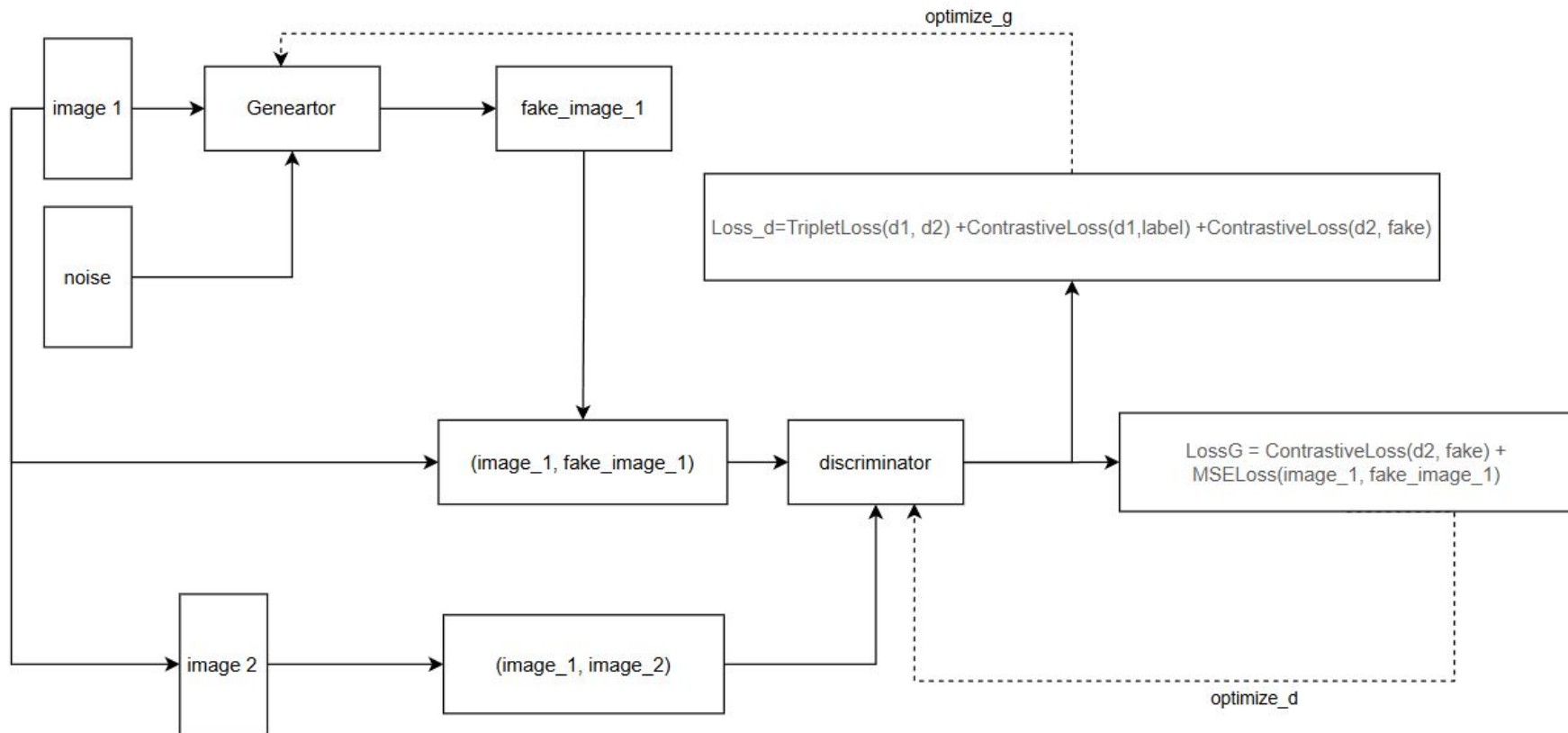
Generator: Unet



Discriminator: Siamese model



Train



Hàm loss

Sử dụng 3 loại loss kết hợp với nhau:

- Contrastive Loss

$$\text{ContrastiveLoss}(d, \text{label}) = (1 - \text{label}) \cdot d^2 + \text{label} \cdot (\max(0, \text{margin} - d))^2$$

- Triplet Loss

$$\text{TripletLoss}(\text{pos_dist}, \text{neg_dist}) = \max(0, \text{pos_dist} - \text{neg_dist} + \text{margin})$$

- MSE Loss

Loss Discriminator

$d(\text{image_1}, \text{image_2}) = d1$

$d(\text{image_1}, \text{image_2}) = d2$

$\text{LossD} = \text{TripletLoss}(d1, d2) + \text{ContrastiveLoss}(d1, \text{label}) + \text{ContrastiveLoss}(d2, \text{fake})$

Loss Generator

$d(\text{image_1}, \text{image_2}) = d1$

$d(\text{image_1}, \text{image_2}) = d2$

$\text{LossG} = \text{ContrastiveLoss}(d2, \text{fake}) + \text{MSELoss}(\text{image_1}, \text{fake_image_1})$

Dataset

- CEDAR
- Hindi
- Bengali

Thiết lập dataset

- Chia cặp:

(genuine, genuine) = số chữ kí thật C_2 của 1 user

(genuine, forged) = số chữ kí thật x số chữ kí giả của 1 user

- Cách chia fold:

chia đều user vào các fold, chứa đầy đủ (genuine, genuine) của các user được chia đều vào fold

(genuine, forged) = (genuine, genuine)

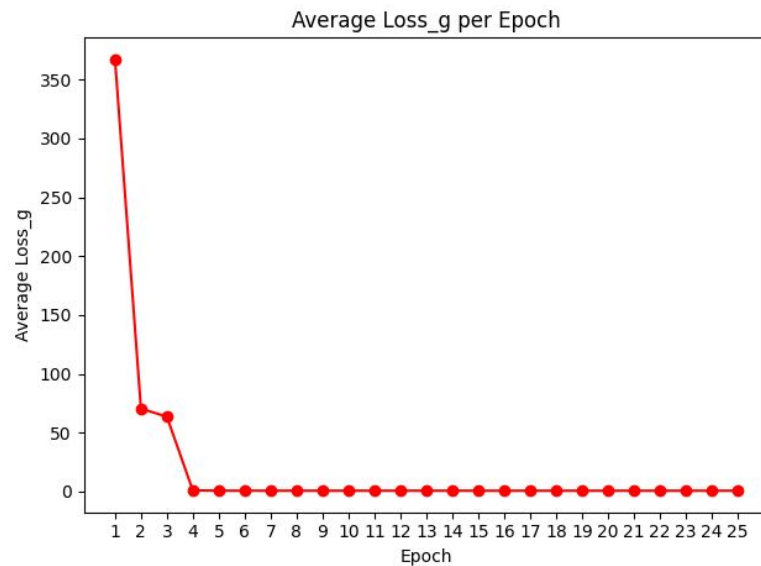
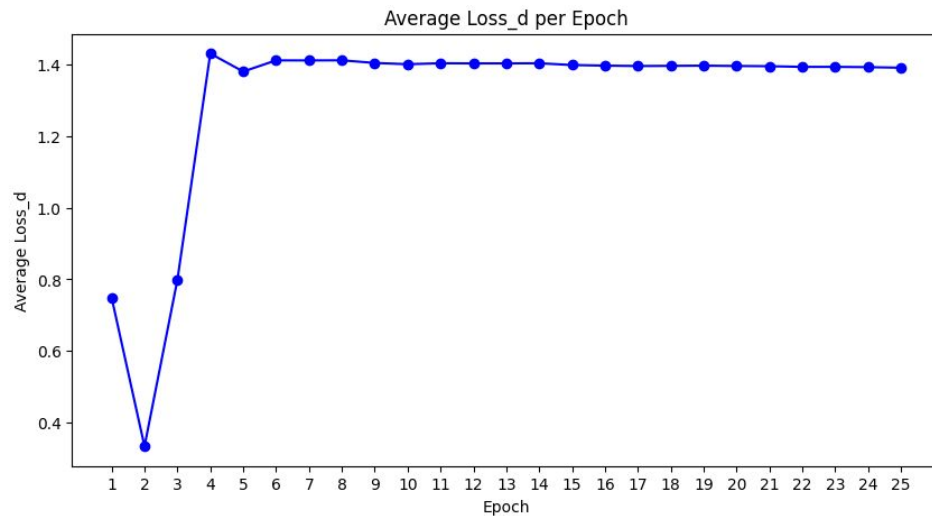
Metric

- FRR, FAR, Acc

		Predicted Class			
		Positive Genuine Signature	Negative Forged Signature		
Actual Class	Positive Genuine Signature	True Positive (TP)	False Negative (FN) Type-I Error	$\text{FRR} = \frac{\text{FN}}{(\text{TP} + \text{FN})}$	$\text{AER} = \frac{\text{FAR} + \text{FRR}}{2}$
	Negative Forged Signature	False Positive (FP) Type-II Error	True Negative (TN)	$\text{FAR} = \frac{\text{FP}}{(\text{FP} + \text{TN})}$	
		$\text{ACC} = \frac{\text{TP} + \text{TN}}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}$		EER When in plot, FRR = FAR	

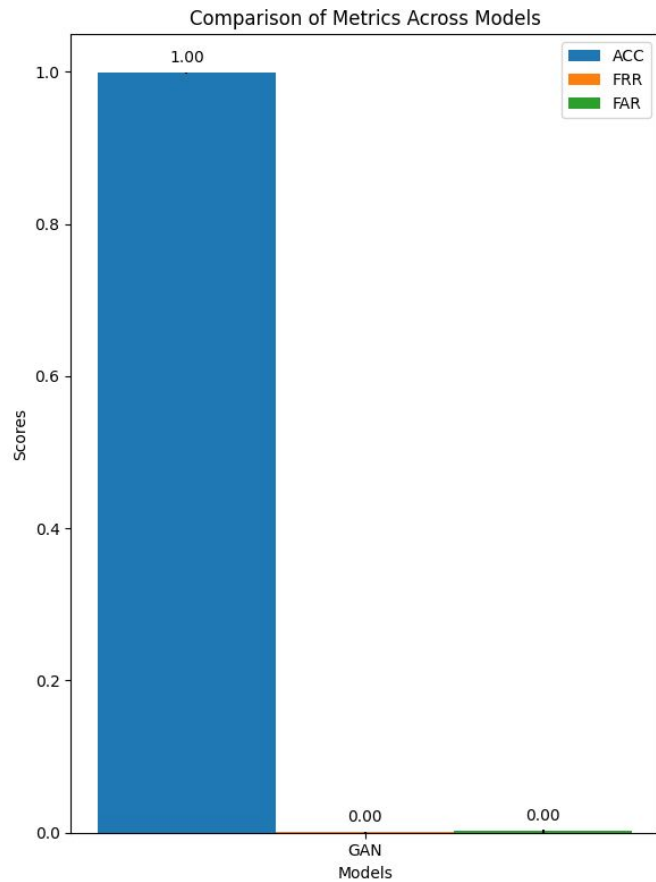
Kết quả

BỘ CEDAR



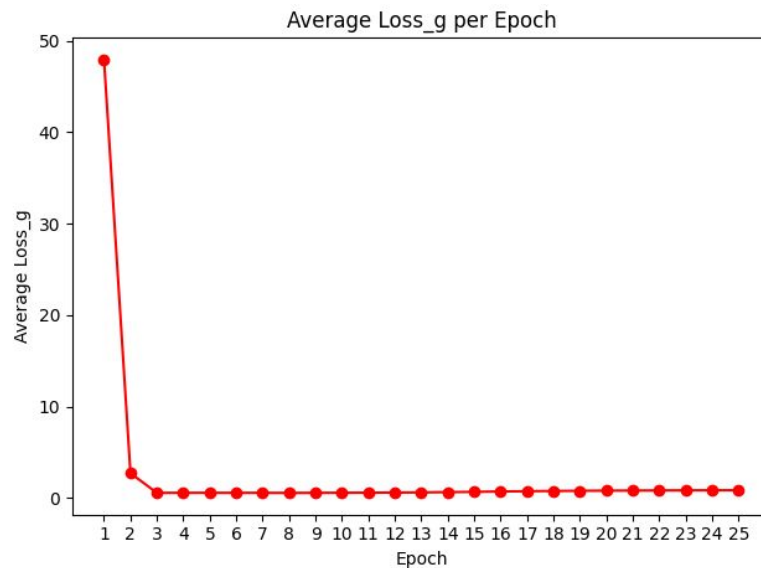
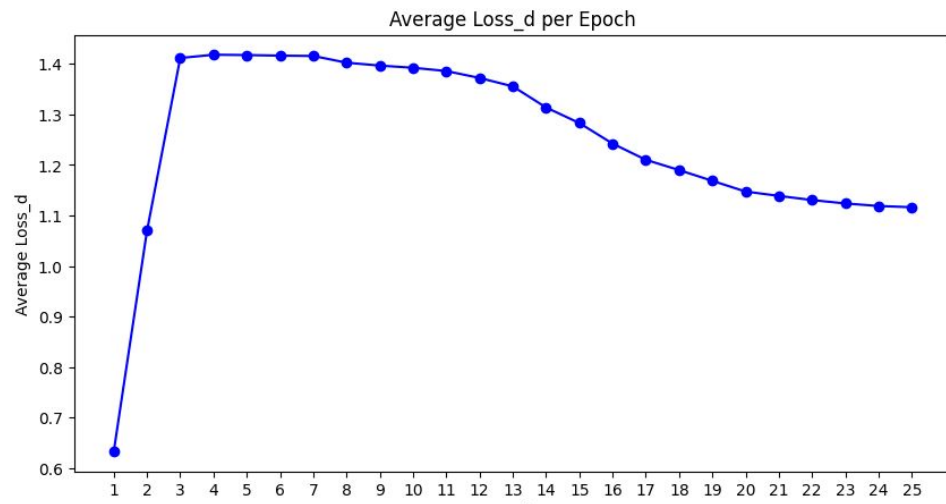
Kết quả

BỘ CEDAR



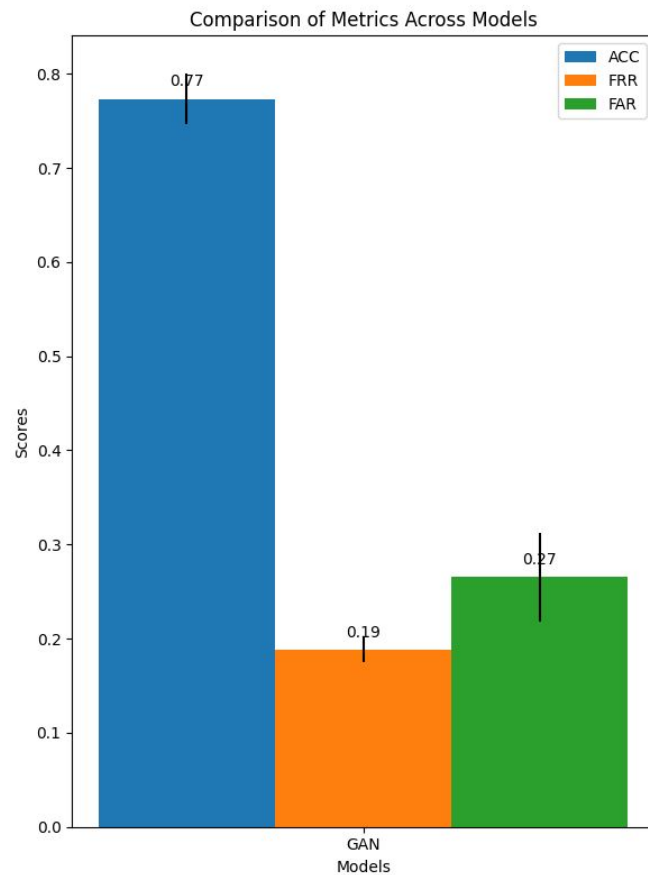
Kết quả

Bộ Hindi



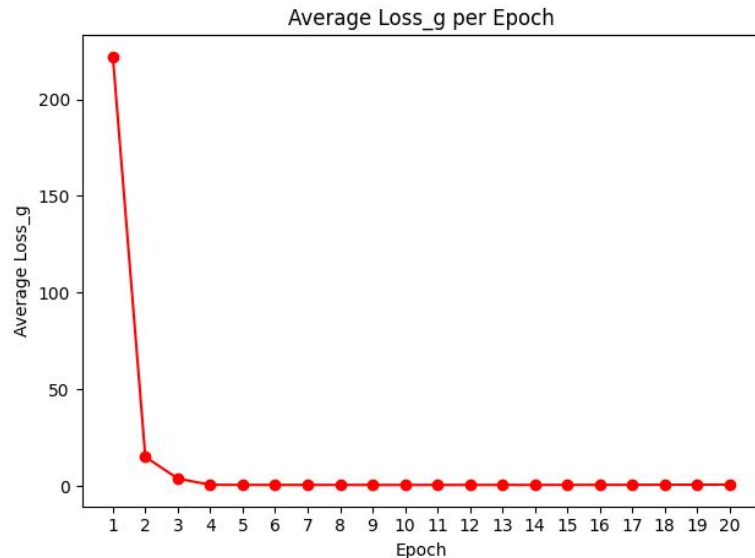
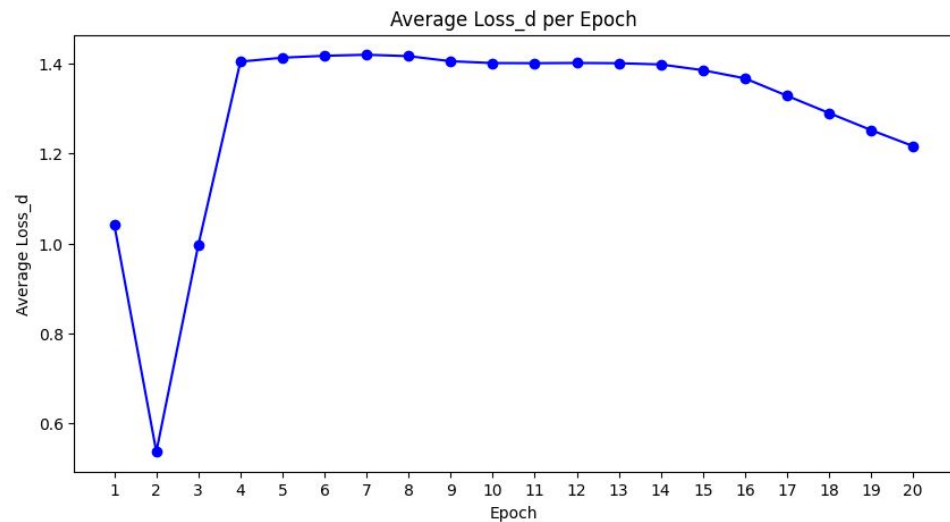
Kết quả

Bộ



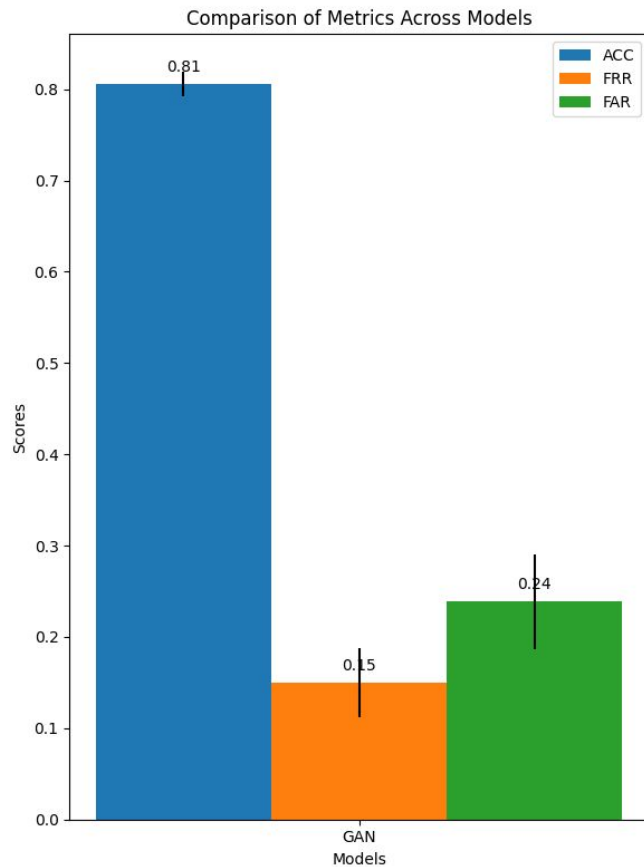
Kết quả

Bộ Bengali



Kết quả

Bộ Bengali



Generator

Nhấn vào để xem ảnh được sinh ra qua từng epochs



Tổng hợp

Dataset	acc	frr	far
CEDAR	0.9987 ± 0.0008	0.0005 ± 0.0004	0.0021 ± 0.0019
Hindi	0.7733 ± 0.0270	0.1882 ± 0.0126	0.2651 ± 0.0467
Bengali	0.8057 ± 0.0137	0.1501 ± 0.0380	0.2386 ± 0.0517

Nhận xét

- Do GAN dựa trên 2 mạng, nên quá trình train khá tốn thời gian.
- Kết quả trên bộ CEDAR khá tốt.
- Có thể thấy mô hình đã hội tụ.
- Nhìn theo hướng khác, thì có thể nói bộ Generator như một kỹ thuật Data Augmentation giúp Discriminator học cách phân loại tốt hơn.

Vấn đề

- Thêm noise vào unet thể nào cho hợp lí, để tạo đa dạng cho nét chữ.
- Hiện tại, generator giống như tạo ra biến thể nhiều của chữ kí đầu vào.
- Nên thiết kế unet như nào để nó giữ lại đặc trưng của chữ kí.
- Loss generator khi có MSE loss vào thì dễ hội tụ, nhưng lại khiến chữ kí giả sinh ra chưa khác lắm với chữ kí thật đầu vào