**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN CUỐI KỲ**

**CHUYÊN ĐỀ NGHIÊN CỨU VÀ ỨNG DỤNG**

**VỀ THỊ GIÁC MÁY TÍNH**

# ỨNG DỤNG SWIN TRANSFORMER TRONG PHÂN LOẠI VẤN ĐỀ CỦA CHẤT LƯỢNG HÌNH ẢNH

**GIẢNG VIÊN HƯỚNG DẪN: TS. Mai Tiến Dũng**

**Thành viên: Vũ Bảo Quốc 220101027**

**Đinh Văn Hoàn 220201030**

**Năm 2023**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN CUỐI KỲ**

**CHUYÊN ĐỀ NGHIÊN CỨU VÀ ỨNG DỤNG**

**VỀ THỊ GIÁC MÁY TÍNH**

# ỨNG DỤNG SWIN TRANSFORMER TRONG PHÂN LOẠI VẤN ĐỀ CỦA CHẤT LƯỢNG HÌNH ẢNH

**GIẢNG VIÊN HƯỚNG DẪN: TS. Mai Tiến Dũng**

**Thành viên: Vũ Bảo Quốc 220101027**

**Đinh Văn Hoàn 220201030**

**Năm 2023**

# MỤC LỤC

# DANH MỤC HÌNH ẢNH

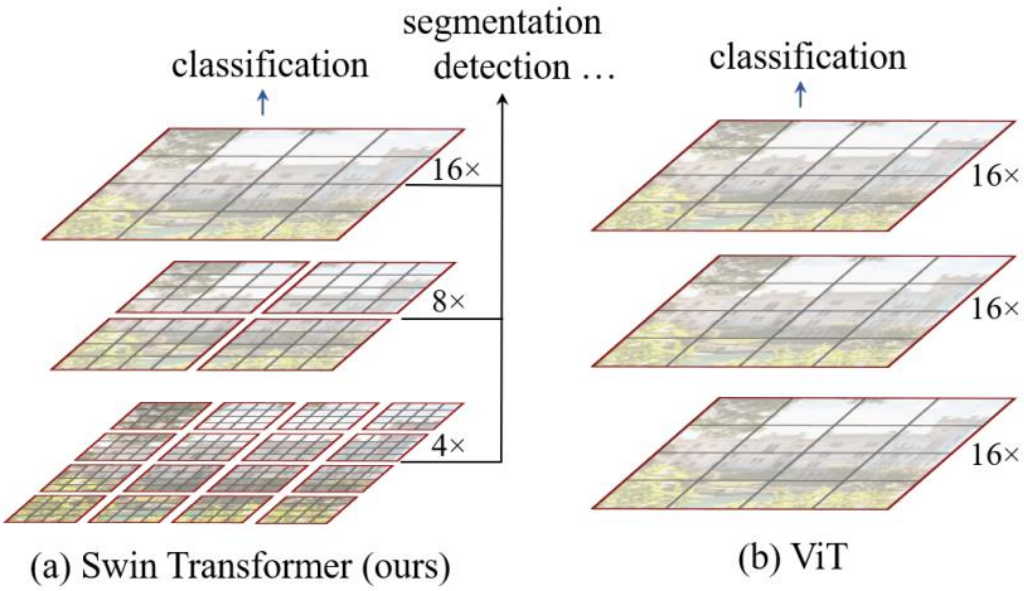
# DANH MỤC BẢNG

# CHƯƠNG 1: TỔNG QUAN

Phân loại vấn đề của hình ảnh là một ý tưởng máy học thú vị, nó giải quyết các thách thức trong thế giới thực trên nhiều ứng dụng khác nhau (Ví dụ: phát hiện và phân loại hình ảnh noisy, ảnh blur trong hệ thống giám sát hoặc tự động kiểm tra chất lượng khi chụp ảnh bằng điện thoại thông minh, v.v.). Chất lượng hình ảnh có thể tác động đáng kể đến kết quả của các nhiệm vụ khác nhau, chi phối tính hiệu quả của các mô hình máy học.

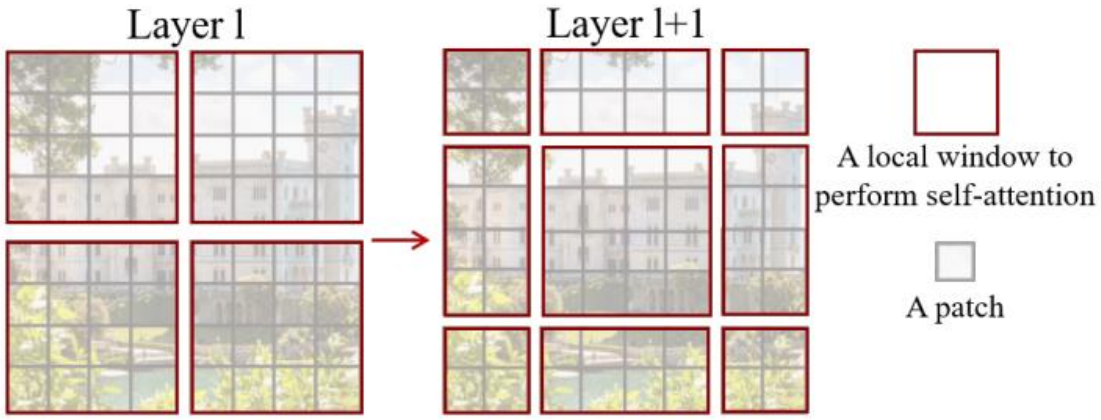
Trong xử lý hình ảnh, các mạng nơ-ron tích chập (CNNs-Convolutional neural networks) hoạt động tốt, chúng có thể khái quát hóa từ dữ liệu quy mô lớn. Tuy nhiên, hoạt động cơ bản trong CNN là "tích chập" cung cấp kết nối cục bộ và tương đương dịch thuật. Mặc dù các thuộc tính này mang lại hiệu quả và khái quát hóa cho CNN, nhưng chúng cũng gây ra hai vấn đề chính là **(1)** Toán tử tích chập có trường tiếp nhận hạn chế, do đó ngăn không cho nó mô hình hóa các phụ thuộc pixel tầm xa. **(2)** Các bộ lọc tích chập có trọng số tĩnh khi suy luận, và do đó không thể thích ứng linh hoạt với nội dung đầu vào. Để giải quyết những thiếu sót nêu trên, một giải pháp thay thế mạnh mẽ và năng động hơn là *self-attention* (SA) tính toán phản hồi tại một pixel nhất định bằng tổng trọng số của tất cả các vị trí khác (Dosovitskiy et al., 2020; Vaswani et al., 2017; Wang et al., 2018; Zhang et al., 2019).

Self-attention là một thành phần cốt lõi trong các mô hình Transformer nhưng với việc triển khai độc đáo, như *multihead SA* giúp tối ưu để song song hóa và học biểu diễn hiệu quả (Khan et al., 2022; Vaswani et al., 2017). Transformer đã cho thấy một SOTA (State-of-the-art) hiệu quả trong xử lý ngôn ngữ tự nhiên và các vấn đề về thị giác máy tính (Brown et al., 2020; Fedus et al., 2022). Mặc dù SA có hiệu quả cao trong việc capturing các tương tác pixel tầm xa, độ phức tạp của nó cũng tăng gấp bốn lần với độ phân giải không gian, do đó không thể áp dụng cho xử lý hình ảnh có độ phân giải cao (Là các trường hợp hổ biến trong phân loại vấn đề chất lượng hình ảnh). Gần đây, một vài nỗ lực đã được thực hiện để điều chỉnh Transformers cho các nhiệm vụ phân loại hình ảnh nhằm giảm tải khối lượng tính toán.




***Hình 1.1. So sánh cách xây dựng feature map cho bài toán image classification và dense recognition giữa Swin Transformer và Vision Transformer*** (Z. Liu et al., 2021). *(A) Swin Transformer xây dựng các classification feature map bằng cách hợp nhất các bản vá hình ảnh (Màu xám) trong các lớp sâu hơn và có độ phức tạp tính toán tuyến tính để nhập kích thước hình ảnh do tính toán SA chỉ trong mỗi cửa sổ cục bộ (Màu đỏ). Do đó, nó có thể là một xương sống cho cả các vấn đề image classification và dense recognition. (B) Ngược lại, các vision Transformers trước đây tạo ra các feature map có độ phân giải thấp duy nhất và có độ phức tạp tính toán bậc hai đối với kích thước hình ảnh đầu vào do tính toán SA trên toàn cục (Brown et al., 2020).*

Swin Transformer (Transformer using Shifted Windows) với kiến trúc điển hình là sự trượt của cửa sổ phân vùng giữa các lớp SA liên tiếp, như minh họa trong **Hình 1.2**. Các Cửa sổ trượt bắc cầu các cửa sổ của lớp trước, cung cấp các kết nối giữa chúng giúp tăng cường đáng kể sức mạnh mô hình hóa (**Bảng 1.1**). Thiết kế này cũng hiệu quả liên quan đến độ trễ trong thực tế: Tất cả các bản vá truy vấn trong một cửa sổ đều chia sẻ cùng một bộ khóa, tạo điều kiện truy cập bộ nhớ trong phần cứng. Ngược lại, các phương pháp SA dựa trên cửa sổ trượt trước đó có độ trễ thấp trên phần cứng chung do các bộ khóa khác nhau cho các pixel truy vấn khác nhau (Hu et al., 2019; Ramachandran et al., 2019). Các nghiên cứu gần đây cũng cho thấy phương pháp Cửa sổ trượt này cũng hiệu quả cho tất cả các kiến trúc MLP (Tolstikhin et al., 2021).




***Hình 1.2. Minh họa về phương pháp Shifted Windows (Cửa sổ trượt) để tính toán SA trong kiến trúc Swin Transformer*** (Z. Liu et al., 2021). *Ở layer 1 (Bên trái), áp dụng một phương pháp phân chia cửa sổ thông thường, và SA được tính toán trong mỗi cửa sổ. Ở layer tiếp theo 1+1 (Bên phải), phân chia cửa sổ được trượt, tạo ra các cửa sổ mới. Việc tính toán SA trong các cửa sổ mới vượt qua ranh giới của các cửa sổ trước đó ở layer 1, tạo ra kết nối giữa chúng.*

***Bảng 1.1. Nghiên cứu tiếp cận phương pháp cửa sổ trượt và các phương pháp nhúng vị trí khác nhau trên 3 benchmark, sử dụng kiến trúc Swin Transformer*** (Z. Liu et al., 2021)**.**

| | ImageNet | | COCO | | ADE20k |
|---|---|---|---|---|---|
| | top-1 | top-5 | $AP^{box}$ | $AP^{mask}$ | mIoU |
| w/o shifting | 80.2 | 95.1 | 47.7 | 41.5 | 43.3 |
| shifted windows | **81.3** | **95.6** | **50.5** | **43.7** | **46.1** |
| no pos. | 80.1 | 94.9 | 49.2 | 42.6 | 43.8 |
| abs. pos. | 80.5 | 95.2 | 49.0 | 42.4 | 43.2 |
| abs.+rel. pos. | 81.3 | 95.6 | 50.2 | 43.4 | 44.0 |
| rel. pos. w/o app. | 79.3 | 94.7 | 48.2 | 41.9 | 44.1 |
| rel. pos. | **81.3** | **95.6** | **50.5** | **43.7** | **46.1** |

Trong đồ án này, nhóm ứng dụng mô hình Swin Transformer để phân loại các vấn đề về chất lượng của hình ảnh bao gồm: Ảnh nhòe (Blurred) defocus và motion, xuất hiện các vệt mưa (Rain), nhiễu (Noisy).

# CHƯƠNG 2: DATASETS VÀ PHƯƠNG PHÁP

## 2.1. Datasets

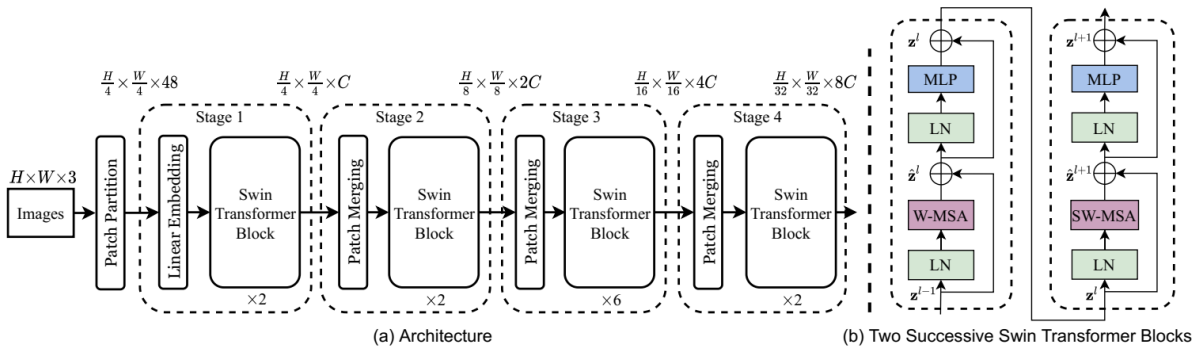
Dataset được sử dụng trong đồ án bao gồm 25.119 ảnh (17.582 ảnh tập train và 7.537 ảnh tập test) được thu thập và tổng hợp từ nhiều dataset khác nhau. Trong đó, data defocused-blurred được thu thập từ DPDD (Dual-Pixel Defocus Deblurring) (Abuolaim & Brown, 2020). Motion-blured dataset được thu thập từ 4 bộ dataset bao gồm GoPro, HIDE, RealBlurR và RealBlurJ_test (Nah et al., 2017; Rim et al., 2020; Shen et al., 2019). Noise images dataset được thu thập từ dataset SIDD (Smartphone-image-denoising-dataset) (Abdelhamed et al., 2018). Rain images dataset được thu thập từ tập Rain13K trong dataset tổng hợp Synthetic rain datasets (Jiang et al., 2020).

***Bảng 2.1. Chi tiết về dataset collection***

| Dataset | Số lượng ảnh (train/test) | Nguồn thu thập |
|---|---|---|
| **Defocused-blurred** | 734 (734/315) | DPDD (Dual-Pixel Defocus Deblurring) |
| **Motion-blurred** | 5.199 (5.039/2.160) | GoPro, HIDE, RealBlurR, RealBlurJ_test |
| **Noise images** | 3,160 (2.212/948) | SIDD (Smartphone-image-denoising-dataset) |
| **Rain images** | 13,711 (9.597/4.114) | Synthetic rain datasets (tập Rain13K) |

## 2.2. Kiến trúc tổng quan của Swin Transformer

Tổng quan bức tranh kiến trúc Swin Transformer (**Hình 2.1**), minh họa cho phiên bản nhỏ (SwinT). Đầu tiên, nó chia một hình ảnh RGB đầu vào thành các mảng không chồng chéo bằng một mô-đun chia patch, tương tự như trong ViT. Mỗi mảng được xem như một "token" và đặc trưng của nó được thiết lập là sự kết hợp của các giá trị RGB nguyên bản của pixel. Kích thước patch là 4×4, vì vậy kích thước đặc trưng của mỗi patch là 4×4×3 = 48. Một lớp nhúng tuyến tính được áp dụng lên đặc trưng này có giá trị nguyên để ánh xạ lên một chiều bất kỳ (Được ký hiệu là C).

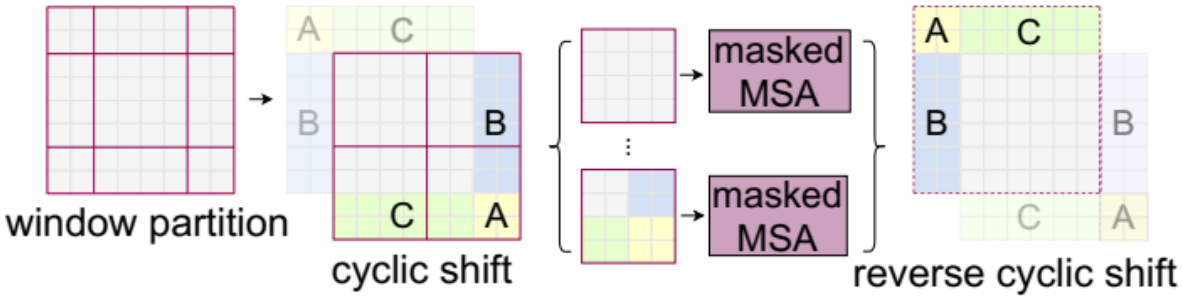



***Hình 2.1. Tổng quan kiến trúc của Swin Transformer*** (Z. Liu et al., 2021)***.*** *(A) Kiến trúc của một Swin Transformer (Swin-T); (B) Hai khối Swin Transformer liên tiếp. W-*

*MSA và SW-MSA là các mô-đun multi-head self attention với cấu hình cửa sổ thông thường và cửa sổ trượt tương ứng.*

Nhiều khối Transformer với tính toán SA được sửa đổi (Block Swin Transformer) được áp dụng lên những token patch này. Các block Transformer giữ số lượng token $\left(\frac{H}{4} \times \frac{W}{4}\right)$, và cùng với lớp nhúng tuyến tính được gọi là "Stage 1". Để tạo ra một biểu diễn phân cấp, số lượng token được giảm bớt thông qua các lớp gộp patch khi mạng sâu hơn. Lớp gộp patch đầu tiên kết hợp các đặc trưng của mỗi nhóm 2×2 patch lân cận và áp dụng một lớp tuyến tính lên các đặc trưng nối 4C chiều. Điều này giảm số lượng token đi một bội số của 2×2=4 (Gảm độ phân giải xuống 2 lần), và kích thước đầu ra được thiết lập là 2C. Khối Swin Transformer được áp dụng sau đó để biến đổi đặc trưng, với độ phân giải giữ nguyên là $\left(\frac{H}{8} \times \frac{W}{8}\right)$. Khối gộp patch và biến đổi đặc trưng đầu tiên này được ký hiệu là "Stage 2". Quy trình này được lặp lại hai lần, với "Stage 3" và "Stage 4", với độ phân giải đầu ra là $\left(\frac{H}{16} \times \frac{W}{16}\right)$ và $\left(\frac{H}{32} \times \frac{W}{32}\right)$ tương ứng. Những giai đoạn này cùng tạo ra một biểu diễn phân cấp, với các đặc trưng giống như các mạng tích chập thông thường, ví dụ như VGG và ResNet (Hendrycks & Gimpel, 2016; X. Liu et al., 2019). Do đó, kiến trúc đề xuất có thể dễ dàng thay thế mạng nền trong các phương pháp hiện tại cho nhiều bài toán của thị giác máy tính.

Block Swin Transformer được xây dựng bằng cách thay thế mô-đun multi-head self attention (MSA) tiêu chuẩn trong một khối Transformer bằng một mô-đun dựa trên cửa sổ trượt, với các lớp khác giữ nguyên. Như minh họa trong **Hình 2.1(B)**, một khối Swin Transformer bao gồm một mô-đun MSA dựa trên cửa sổ trượt, tiếp theo là một MLP 2 lớp với không gian giữa là hàm active GELU. Một lớp LayerNorm (LN) được áp dụng trước mỗi mô-đun MSA và mỗi MLP, và một kết nối dư được áp dụng sau mỗi mô-đun.




***Hình 2.2. Minh họa về một phương pháp tính toán batch hiệu quả cho SA trong phân chia cửa sổ trượt*** (Z. Liu et al., 2021)**.**

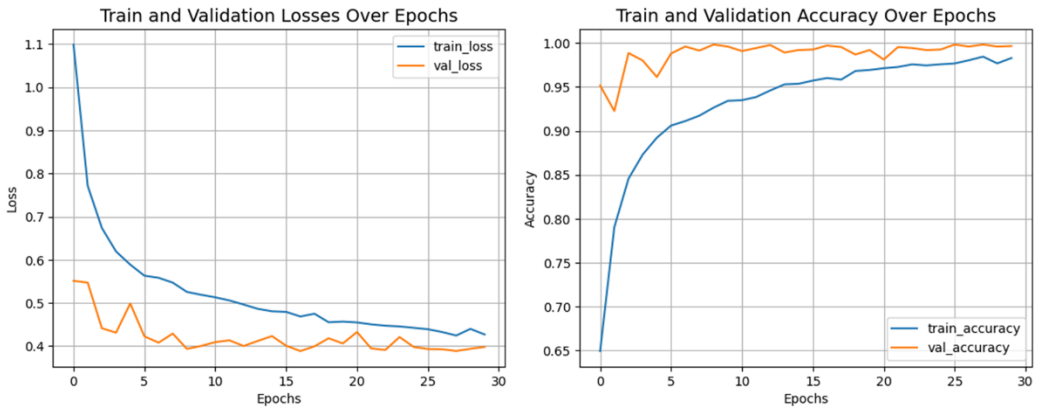
# CHƯƠNG 3: KẾT QUẢ

## 3.1. Training model

Kết quả training model được ghi lại mỗi epoch, với thông tin về loss (mất mát), accuracy (độ chính xác), trên tập huấn luyện và tập kiểm thử. Đối với epoch đầu tiên, mô hình có loss trung bình trên tập huấn luyện là 1.0987, độ chính xác là 64.95%, và độ chính xác trong top 5 là 100%. Sau 30 epoch, mô hình có loss trung bình trên tập kiểm thử là 0.3981, độ chính xác là 99.66%, và độ chính xác trong top 5 là 100%. Kết quả này cho thấy mô hình đã học được một cách hiệu quả và có khả năng dự đoán chính xác trên cả tập huấn luyện và tập kiểm thử.

***Bảng 3.1. Kết quả Huấn luyện Mô hình qua 30 epoch***

| Epoch | Loss | Accuracy | Val Loss | Val Accuracy |
|---|---|---|---|---|
| 1 | 1.0987 | 0.6495 | 0.5511 | 0.9517 |
| 2 | 0.7714 | 0.7905 | 0.5471 | 0.9227 |
| 3 | 0.6737 | 0.8457 | 0.4414 | 0.9886 |
| 4 | 0.6192 | 0.8731 | 0.431 | 0.9801 |
| 5 | 0.5888 | 0.8923 | 0.4987 | 0.9613 |
| 6 | 0.5631 | 0.9061 | 0.4226 | 0.9881 |
| 7 | 0.5583 | 0.9112 | 0.4079 | 0.996 |
| 8 | 0.547 | 0.9174 | 0.429 | 0.9915 |
| 9 | 0.5255 | 0.9265 | 0.3935 | 0.9983 |
| 10 | 0.5189 | 0.9342 | 0.4002 | 0.996 |
| 11 | 0.513 | 0.935 | 0.4092 | 0.9909 |
| 12 | 0.5057 | 0.9386 | 0.4134 | 0.9943 |
| 13 | 0.4963 | 0.9461 | 0.4001 | 0.9977 |
| 14 | 0.4863 | 0.953 | 0.4119 | 0.9892 |
| 15 | 0.4806 | 0.9537 | 0.4232 | 0.992 |
| 16 | 0.4793 | 0.9573 | 0.4009 | 0.9926 |
| 17 | 0.4686 | 0.9602 | 0.3884 | 0.9972 |
| 18 | 0.475 | 0.9584 | 0.3994 | 0.9955 |
| 19 | 0.4552 | 0.9682 | 0.4181 | 0.9869 |
| 20 | 0.4567 | 0.9693 | 0.4061 | 0.992 |
| 21 | 0.4548 | 0.9714 | 0.4325 | 0.9812 |
| 22 | 0.4503 | 0.9728 | 0.3945 | 0.9955 |
| 23 | 0.4472 | 0.9757 | 0.391 | 0.9943 |
| 24 | 0.4455 | 0.9745 | 0.421 | 0.992 |
| 25 | 0.4423 | 0.9758 | 0.3979 | 0.9926 |
| 26 | 0.439 | 0.9768 | 0.3932 | 0.9983 |
| 27 | 0.4326 | 0.9804 | 0.3926 | 0.996 |
| 28 | 0.4245 | 0.9845 | 0.3886 | 0.9983 |
| 29 | 0.4398 | 0.9769 | 0.3937 | 0.996 |
| 30 | 0.4269 | 0.9829 | 0.3981 | 0.9966 |

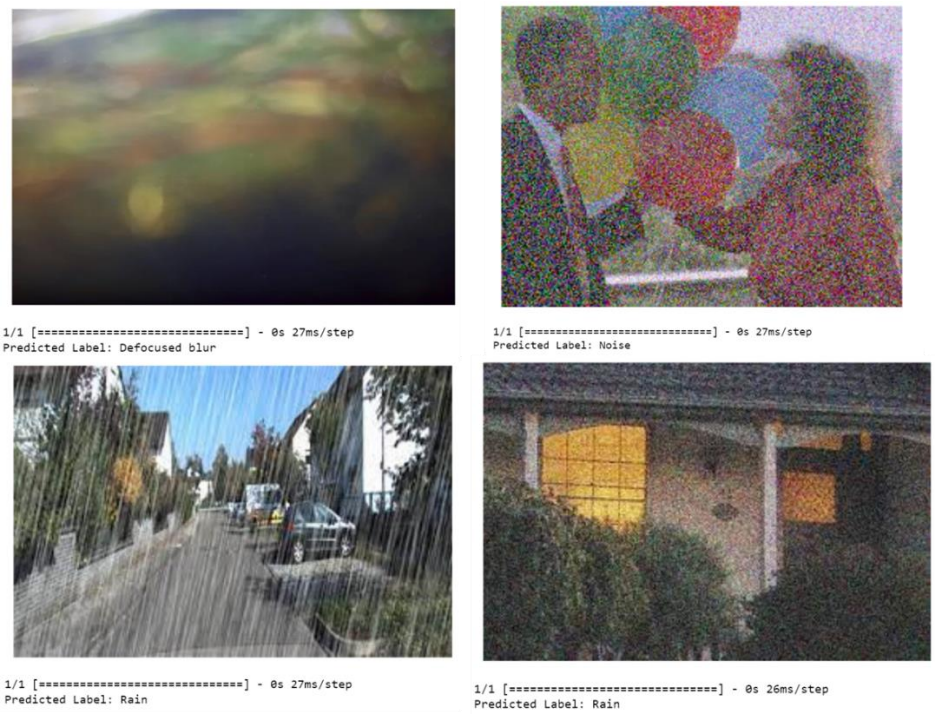



***Hình 3.1. Biểu Đồ kết quả huấn luyện và kiểm thử của Loss và Accuracy qua Các Epochs***

### 3.2. Testing model

Kết quả kiểm thử của mô hình sau 236 bước thử nghiệm đã đạt được hiệu suất cao trên tập kiểm thử, có loss = 0.4814 với độ chính xác đạt 95.49%. Kết quả này cho thấy mô hình có khả năng phân loại tương đối chính xác với dữ liệu test. Đây là một kết quả tích cực, chứng minh rằng mô hình đã học được cách đại diện cho dữ liệu và thực hiện dự đoán chính xác trên tập test.

Kết quả thử nghiệm ngẫu nhiên trên 4 ảnh được thu thập ngẫu nhiên trên internet (**Hình 3.2**). Trong đó ở kết quả cuối cùng giữa noise và rain mô hình đã phân loại sai từ noise thành rain. Điều này có thể do sự không đồng đều trong dataset noise và do trong tập train và tập test thì cả 2 bộ dataset này đều có phần lớn là ảnh được tạo ra từ các thuật toán.




***Hình 3.2. Kết quả thử nghiệm ngẫu nhiên trên 4 ảnh được thu thập ngẫu nhiên trên internet.***

# CHƯƠNG 4: ĐÁNH GIÁ VÀ KẾT LUẬN

## 4.1. Kết luận

Kết quả thử nghiệm của mô hình Swin Transformer trong việc phân loại các vấn đề chất lượng hình ảnh cho thấy mô hình đã đạt được hiệu suất cao trong quá trình huấn luyện, với độ chính xác và độ tin cậy tăng lên đáng kể qua các epoch. Điều này chỉ ra khả năng học tốt của Swin Transformer trên dữ liệu chất lượng hình ảnh. Bên cạnh đó kết quả kiểm thử cho thấy mô hình có khả năng tổng quát hóa tốt trên dữ liệu mới, với độ chính xác ổn định và loss thấp. Điều này là quan trọng để đảm bảo tính ứng dụng của mô hình trong các tình huống thực tế. Mặc dù có hiệu suất cao, mô hình vẫn mắc phải một số sai sót trong việc phân loại ảnh từ dữ liệu ngẫu nhiên thu thập từ internet. Điều này có thể đến từ sự không đồng đều trong dataset và tính ngẫu nhiên của dữ liệu kiểm thử.

## 4.2. Kiến Nghị

Để cải thiện khả năng phân loại ảnh trong các trường hợp đặc biệt, như phân loại ảnh noise và rain, cần tiếp tục tối ưu hóa và đào tạo mô hình với các trường hợp này để giảm thiểu sai sót.

Mở rộng dataset với độ đa dạng cao và thực hiện các kỹ thuật tăng cường dữ liệu sẽ giúp mô hình hiểu biến thể lớn trong dữ liệu, làm tăng khả năng tổng quát hóa.

Đối với việc xử lý hình ảnh có độ phân giải cao, cần xem xét các kỹ thuật tối ưu hóa tính toán để giảm độ phức tạp của mô hình Swin Transformer và làm cho nó thích hợp cho các ứng dụng thực tế.

Kiểm soát overfitting, chẳng hạn như sử dụng các kỹ thuật dropout hoặc regularization, có thể giúp đảm bảo rằng mô hình không chỉ học thuộc lòng dữ liệu huấn luyện mà còn có khả năng tổng quát hóa tốt trên dữ liệu mới.

# TÀI LIỆU THAM KHẢO


Abdelhamed, A., Lin, S., & Brown, M. S. (2018). A high-quality denoising dataset for smartphone cameras. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1692–1700.

Abuolaim, A., & Brown, M. S. (2020). Defocus deblurring using dual-pixel data. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, 111–126.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., & Askell, A. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., & Gelly, S. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv Preprint ArXiv:2010.11929*.

Fedus, W., Zoph, B., & Shazeer, N. (2022). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, *23*(1), 5232–5270.

Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). *ArXiv Preprint ArXiv:1606.08415*.

Hu, H., Zhang, Z., Xie, Z., & Lin, S. (2019). Local relation networks for image recognition. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3464–3473.

Jiang, K., Wang, Z., Yi, P., Chen, C., Huang, B., Luo, Y., Ma, J., & Jiang, J. (2020). Multi-scale progressive fusion network for single image deraining. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8346–8355.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*, *54*(10s), 1–41.

Liu, X., Suganuma, M., Sun, Z., & Okatani, T. (2019). Dual residual networks leveraging the potential of paired operations for image restoration. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7007–7016.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.

Nah, S., Hyun Kim, T., & Mu Lee, K. (2017). Deep multi-scale convolutional neural network for dynamic scene deblurring. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3883–3891.

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., & Shlens, J. (2019). Stand-alone self-attention in vision models. *Advances in Neural Information Processing Systems*, *32*.

Rim, J., Lee, H., Won, J., & Cho, S. (2020). Real-world blur dataset for learning and benchmarking deblurring algorithms. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, 184–201.

Shen, Z., Wang, W., Lu, X., Shen, J., Ling, H., Xu, T., & Shao, L. (2019). Human-aware motion deblurring. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5572–5581.

Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., & Uszkoreit, J. (2021). Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, *34*, 24261–24272.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7794–7803.

Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019). Self-attention generative adversarial networks. *International Conference on Machine Learning*, 7354–7363.