TRƯỜNG ĐẠI HỌC KHOA HỌC
KHOA CÔNG NGHỆ THÔNG TIN

SỐ PHÁCH:

NHẬN DẠNG TIỀN VIỆT NAM BẰNG MÔ HÌNH VGG16 QUA CAMERA

LÝ THUYẾT NHẬN DẠNG -
2024-2025.2.TIN4243.001
NGUYỄN ĐĂNG BÌNH

HUẾ, THÁNG 6 NĂM 2025

TRƯỜNG ĐẠI HỌC KHOA HỌC
KHOA CÔNG NGHỆ THÔNG TIN

SỐ PHÁCH:

NHẬN DẠNG TIỀN VIỆT NAM BẰNG MÔ HÌNH VGG16 QUA CAMERA

LÝ THUYẾT NHẬN DẠNG - 2024-2025.2.TIN4243.001

Giảng viên hướng dẫn: Nguyễn Đăng Bình

Sinh viên thực hiện: Hồ Trọng Nghĩa

Mã sinh viên: 22T1020683

HUẾ, THÁNG 6 NĂM 2025

MỤC LỤC

MỞ ĐẦU

Kể từ khi tiền tệ ra đời và trở thành phương tiện trao đổi hàng hóa chủ yếu, việc nhận dạng và phân biệt tiền đã trở thành một nhu cầu thiết yếu. Trong bối cảnh phát triển ngày nay, các công nghệ tự động hóa ngày càng được ứng dụng rộng rãi để hỗ trợ con người xử lý các tác vụ lặp đi lặp lại một cách hiệu quả.

Tại Việt Nam, mặc dù nhu cầu nhận dạng tiền tệ tự động ngày càng tăng cao trong các ứng dụng như máy ATM, máy đếm tiền, hệ thống thanh toán tự động, nhưng các nghiên cứu trong lĩnh vực này chủ yếu dựa vào phương pháp truyền thống với việc xác định các đặc trưng thủ công bằng sự quan sát và phân tích của con người. Những phương pháp này thường gặp khó khăn trong việc đảm bảo độ chính xác cao và khả năng thích ứng với các điều kiện môi trường khác nhau, từ đó tạo ra hạn chế đáng kể trong việc ứng dụng thực tế.

Mục tiêu chính của nghiên cứu này là xây dựng một hệ thống nhận dạng tự động các mệnh giá tiền Việt Nam thông qua camera sử dụng mô hình học sâu VGG16. Nghiên cứu áp dụng dụng phương pháp học chuyển giao (transfer learning) từ mô hình VGG16 đã được huấn luyện trước trên tập dữ liệu ImageNet [2], sau đó tiến hành tinh chỉnh (fine-tuning) các tham số để phù hợp với các đặc thù của bài toán nhận dạng tiền Việt Nam.

Cụ thể, hệ thống sẽ có khả năng:

- Phân loại chính xác các mệnh giá tiền Việt Nam từ hình ảnh thu thập qua camera.
- Hoạt động trong thời gian thực với độ chính xác cao.
- Tận dụng ưu điểm của mô hình học sâu để tự động trích xuất đặc trưng mà không cần can thiệp thủ công.

Nghiên cứu này tập trung vào việc phân loại các mệnh giá tiền Việt Nam hiện hành, bao gồm cả tiền giấy cotton và tiền polymer, thông qua dữ liệu hình ảnh được thu thập từ webcam laptop thông thường. Hệ thống sẽ được huấn luyện và kiểm thử trên các mệnh giá: 500,000 VNĐ, 200,000 VNĐ, 100,000 VNĐ, 50,000 VNĐ, 10,000 VNĐ, 5.000 VNĐ, 2,000 VNĐ, 1,000 VNĐ.
Nghiên cứu có những giới hạn nhất định:

- Chỉ thực hiện nhận dạng trong điều kiện ánh sáng tự nhiên và ánh sáng đèn LED thông thường.
- Yêu cầu tờ tiền phải nằm trong khung hình một cách tương đối rõ ràng và không bị che khuất quá 30

- Chỉ nhận dạng tiền Việt Nam trong tình trạng còn sử dụng được (không bao gồm tiền bị rách nát, ẩm ướt nghiêm trọng).
- Sử dụng webcam thông thường với độ phân giải từ 720p trở lên.
- Khoảng cách chụp từ 10-50cm để đảm bảo chất lượng hình ảnh đầu vào.

Những giới hạn này được thiết lập nhằm tập trung vào việc xây dựng một hệ thống cơ bản có thể hoạt động hiệu quả trong điều kiện thực tế phổ biến, đồng thời tạo tiền đề cho các nghiên cứu mở rộng trong tương lai.

Chương 1: TỔNG QUAN

1.1 Bài toán nhận dạng tiền tệ và ứng dụng thực tiễn

Nhận dạng tiền tệ tự động là nhu cầu thiết yếu trong thời đại công nghệ hiện nay. Để hệ thống có tự động nhận biết được mệnh giá tiền. Trước tiên, nó cần thu nhận hình ảnh tờ tiền thông qua camera. Tiếp theo, hệ thống phải phân tích và nhận diện được đặc trưng riêng biệt của từng mệnh giá từ hình ảnh hoặc camera thu thập. Dựa trên những đặc trưng này, hệ thống mới có khả năng phân biệt chính xác mệnh giá của từng tờ tiền.

Để xây dựng một mô hình tự động phân tích và nhận diện đặc trưng, chúng ta cần áp dụng phương pháp học sâu với kiến trúc gồm nhiều lớp nơ-ron dày đặc. Nhờ đó, mô hình có thể tự động trích xuất các đặc trưng quan trọng từ hình ảnh, từ đó phân loại chính xác mệnh giá tiền dựa trên dữ liệu hình ảnh đầu vào.

Bài toán nhận dạng tiền tệ mang lại giá trị thực tiễn cao, được ứng dụng rỗng rãi trong nhiều lĩnh vực:

Bảng 1.1. Tổng hợp các ứng dụng sử dụng công nghệ nhận dạng tiền tệ

Ứng dụng	Mô tả chi tiết
Máy rút tiền tự động (ATM)	Sử dụng công nghệ nhận dạng để kiểm tra và phát ra các tờ tiền chính xác, đảm bảo giao dịch an toàn và hiệu quả.
Máy đếm tiền	Sắp xếp và đếm các tờ tiền theo mệnh giá, giảm thiểu sai sót và tiết kiệm thời gian trong xử lý tiền mặt.
Hệ thống thanh toán tự động	Áp dụng trong máy bán hàng, máy đỗ xe, hoặc phương tiện giao thông công cộng, cung cấp sự tiện lợi cho người dùng.
Ngân hàng và tài chính	Hỗ trợ xử lý các giao dịch như gửi, rút tiền, và các hoạt động tài chính khác, tăng cường độ chính xác.
Bán lẻ và thương mại	Trong máy tính tiền và hệ thống bán hàng, giúp xử lý thanh toán bằng tiền mặt nhanh chóng và đáng tin cậy.

1.2 Các nghiên cứu liên quan:

Trong nghiên cứu của P. Divya Jenifar và các công sự [5], (2023), nhằm nhận diện tiền tệ thông qua hình ảnh, đã sử dụng các phương pháp học sâu dựa trên thị giác máy tính như ResNet, AlexNet để xem mô hình được tạo ra từ kiến trúc nào tốt hơn cho bìa toán nhận diện tiền Ấn Độ. Các mô hình trên điều được áp dụng phương pháp học chuyển giao cả. Hệ thống ResNet trích xuất các đặc trưng sâu từ hình ảnh tốt hơn là sử dụng AlexNet, độ chính xác (accuracy) hệ thông là 96,07%.

César G. Pachón và các công sự [1], (2021), đã đề xuất phương pháp đóng băng lớp, là phương pháp đóng băng các tham số theo một cách thức nhất định. Đối với từng mô hình, người ta tạo ra ba trường hợp đóng băng, là không đóng băng điểm, đóng băng nửa đầu tham số và tất cả tham số đều đóng băng, ngoại trừ các tham số của lớp kết nối đầy đủ cuối cùng. Các tác giả thử nghiệm phương pháp trên qua từng mô hình ResNet18, InceptionV3, SqueezeNet, AlexNet. Ngoài ra, họ còn tạo ra một mô hình tùy chỉnh CNN để giải quyết bài toán nhận dạng tiền này. Hiệu suất của mô hình cho biết ResNet18 đạt độ chính xác (accuracy) cao nhất 100%, các mô hình khác cũng bám sát nút. Có điều mô hình nhẹ và nhanh nhất lại là mô hình CNN tùy chỉnh, với 7,645,125 tham số và 0.145 giây CPU, 0.0054 giây trên GPU.

Muhammad Sarfraz [4], (2015), đã đề xuất một hệ thống nhận diện tiền giấy tự động, sử dụng Radial Basis Function Network cho bài toán này. Khi ảnh cần nhận dạng được nhập vào, hệ thống sẽ tính toán để trích xuất các đặc trưng, sau đó tính hệ số tương quan với các ảnh mẫu, rồi xây dựng xây dựng và huấn luyện mạng RBFN ngay tại thời điểm đó bằng các đặc trưng ảnh mẫu và ảnh cần nhận dạng, thực hiện phân loại mệnh giá tiền. Kết quả của mô hình cho ra độ chính xác sau: 95,37% đối với ảnh bình thường, không nghiên, 91,65% đối với ảnh nhiễu, không nghiên, 87,5% đối với ảnh bị nghiên góc nhỏ hơn 15%. Từ đó, mô hình có tỷ lệ nhận dạng trung bình cho toàn bộ ảnh là 91,51%.

Bảng 1.2. Tổng kết một số nghiên cứu nhận dạng tiền tệ bằng học sâu

Tác giả	Thuật toán / Mô hình	Tập dữ liệu	Độ chính xác (%)
P. Divya Jenifar và cs. (2023)	ResNet, AlexNet (Transfer Learning)	Tiền Ấn Độ (ảnh camera)	96.07
César G. Pachón và cs. (2021)	ResNet18, InceptionV3, SqueezeNet, AlexNet, CNN tùy chỉnh	Tiền giả/nghi ngờ (quốc tế)	100 (ResNet18)
Muhammad Sarfraz (2015)	RBFN (Radial Basis Function Network)	Tiền giấy đa điều kiện ảnh	91.51 (trung bình)

1.3 Thách thức và khoảng trống nghiên cứu:

Mặc dù đã có nhiều nghiên cứu trên thế giới ứng dụng các kỹ thuật học sâu vào bài toán nhận dạng tiền tệ, phần lớn các công trình tập trung vào tiền giấy của các quốc gia như Ấn Độ, Mỹ, Trung Quốc,... Trong khi đó, các nghiên cứu về nhận dạng tiền Việt Nam bằng công nghệ học sâu, đặc biệt là qua camera thời gian thực, vẫn còn hạn chế và chưa phổ biến.

Bên cạnh đó, các thách thức kỹ thuật của bài toán bao gồm:

- Đặc điểm thiết kế của tiền Việt Nam có nhiều chi tiết nhỏ, dễ bị nhầm lẫn giữa các mệnh giá, đặc biệt là giữa 1.000 VNĐ và 2.000 VNĐ do bố cục và màu sắc tương đối giống nhau.
- Điều kiện môi trường (ánh sáng yếu, góc chụp nghiêng, tiền bị nhàu hoặc che khuất một phần) ảnh hưởng lớn đến chất lượng hình ảnh đầu vào.
- Thiếu bộ dữ liệu huấn luyện chuyên biệt cho tiền Việt Nam với đủ độ đa dạng về góc độ, ánh sáng và trạng thái vật lý của tờ tiền.
- Các hệ thống truyền thống yêu cầu xử lý đặc trưng thủ công, thiếu khả năng tự động học đặc trưng từ dữ liệu đầu vào như trong các mô hình học sâu hiện đại.

Chính những khoảng trống và thách thức này cho thấy nhu cầu cấp thiết của việc xây dựng và thử nghiệm một hệ thống nhận dạng tiền Việt Nam sử dụng mô hình học sâu hiện đại như VGG16.

1.4 Đóng góp của bài tiểu luận:

Bài tiểu luận này đóng vai trò như một nghiên cứu thử nghiệm nhằm kiểm chứng khả năng áp dụng mô hình học sâu VGG16 trong bài toán nhận dạng tiền Việt Nam từ hình ảnh camera. Những đóng góp chính bao gồm:

- Thiết kế một pipeline nhận dạng tiền Việt Nam qua camera từ khâu thu thập dữ liệu, xử lý ảnh, đến huấn luyện và dự đoán bằng mô hình VGG16 (transfer learning).
- Ứng dụng kỹ thuật học chuyển giao để tận dụng mô hình VGG16 đã huấn luyện trước trên ImageNet, từ đó giảm thời gian huấn luyện và yêu cầu dữ liệu lớn.
- Thử nghiệm thực tế hệ thống trong điều kiện môi trường phổ biến, sử dụng webcam laptop thông thường và ánh sáng tự nhiên hoặc đèn LED.
- Đánh giá độ chính xác của hệ thống phân loại mệnh giá tiền Việt Nam, từ đó xác định tính khả thi của việc triển khai ứng dụng thực tiễn.
- Tạo tiền đề cho các nghiên cứu mở rộng sau này, như nhận dạng tiền giả, nhận dạng trong điều kiện khó khăn hơn (tiền nhàu, rách), hoặc tích hợp hệ thống vào các thiết bị di động và ATM.

Chương 2: LÝ THUYẾT CƠ SỞ

Ở chương này, tôi sẽ giải thích các lý thuyết cơ sở sẽ được sử dụng trong bài toán *"Nhận dạng tiền Việt Nam bằng mô hình VGG16 qua camera"*. Đầu tiên là mô tả kiến trúc CNN, các phép toán được sử dụng trong CNN. Tiếp theo là kiến trúc VGG16 - kiến trúc sẽ được sử dụng trong bài tiểu luận này.

2.1 Convolutional Neural Network (CNN)

Kiến trúc mạng tích chập (Convolutional Neural Network - CNN) là một mô hình học sâu (*deep learning*) được thiết kế để trích xuất các đặc trưng quan trọng từ dữ liệu phi cấu trúc, như là ảnh. Nó được cấu tạo bởi nhiều lớp, bao gồm:

- Lớp tích chập (*convolutional*),
- Lớp tổng hợp (*pooling*),
- Lớp kết nối đầy đủ (*fully connected*).

Kiến trúc mạng tích chập được dùng để giải quyết các tác vụ quan trọng như:

- Phân loại ảnh (*image classification*),
- Phát hiện đối tượng (*object detection*),
- Phân đoạn ảnh (*image segmentation*).

2.1.1 Phép tích chập (Convolutional)

Đây là một phép tính quan trọng trong kiến trúc mạng tích chập, khi nó được dùng để phát hiện các cạnh trong một hình ảnh (cạnh dọc, cạnh ngang, cạnh $45°$, …), góc, hình dạng, kết cấu. Kết quả là một bản đồ đặc trưng mới được tạo ra, những đặc trưng về một khía cạnh nào đó (dựa theo bộ lọc) được biểu diễn nổi bật. Nhờ vậy, bản đồ này tập trung được các thông tin quan trọng nhất của bản đồ đặc trưng trước đó.

Mục đích của phép tích chập:

- Phát hiện các cạnh, góc, hình dạng, kết cấu của ảnh.
- Giữ lại cấu trúc không gian.
- Giảm số lượng tham số đồng thời cũng giảm lượng tính toán. Do không kết nối đầy đủ như mạng kết nối đầy đủ nên số lượng trọng số ít hơn.
- Khả năng chia sẻ trọng số: trọng số của bộ lọc tại một vị trí cũng được dùng cho các vị trí khác.

- Do sử dụng trọng số dùng chung nên mô hình ít bị khít (*overfit*) hơn, tăng khả năng học khái quát.

Thuật toán phép tích chập:
Giả sử:

- Đầu vào: một bản đồ đặc trưng 3D có kích thước $h \times w \times c$
- Bộ lọc: kích thước $k \times k \times c$
- Bước nhảy (*stride*): s
- Vùng đệm (*padding*): p

Lưu ý: bản đồ đặc trưng $h \times w \times 1$ tương ứng với bộ lọc $k \times k \times 1$ và tương tự cho các chiều lớn hơn.
Các bước thực hiện:

1. Di chuyển bộ lọc đi qua bản đồ đặc trưng từ trái sang phải, từ trên xuống dưới với bước nhảy s.
2. Tại mỗi vị trí, tính tổng các tích phần tử tương ứng giữa bộ lọc và vùng ảnh hiện tại.
3. Ghi lại giá trị đó vào bản đồ đặc trưng đầu ra.
4. Lặp lại cho đến khi quét hết vùng ảnh.

Kích thước đầu ra:

$$H' = \left\lfloor \frac{h + 2p - k}{s} \right\rfloor + 1, \quad W' = \left\lfloor \frac{w + 2p - k}{s} \right\rfloor + 1$$

Trong đó:

- H', W': chiều cao và chiều rộng đầu ra
- h, w: kích thước đầu vào
- k: kích thước bộ lọc
- s: bước nhảy
- p: số lượng padding

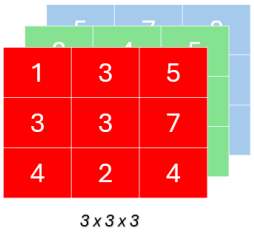
Ví dụ minh họa:
Bước 1: Đầu vào và Bộ lọc Ta có đầu vào là một bản đồ đặc trưng kích thước $3 \times 3 \times 3$ và một bộ lọc có kích thước $2 \times 2 \times 3$ tương ứng.

Bước 2: Thực hiện tích chập tại một vị trí đầu ra
Ta trượt bộ lọc qua một vùng của ảnh đầu vào với stride $s = 1$, thực hiện tích chập cho từng kênh và cộng kết quả lại.
Chú thích ký hiệu:

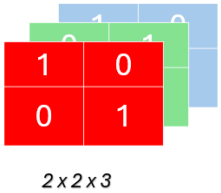
- $a_{h,w,c}$: giá trị tại vị trí (h, w) của kênh màu c trong bản đồ đặc trưng đầu vào
- $f_{h,w,c}$: giá trị tại vị trí (h, w) của kênh c trong bộ lọc
- $c_{h,w,c}$: kết quả tích chập tại vị trí đầu ra (h, w) trên kênh c
- $o_{h,w}$: tổng các giá trị tích chập tại đầu ra (h, w) sau khi cộng tất cả kênh

Bước 3, 4, 5: Dịch chuyển bộ lọc
Tiếp tục dịch bộ lọc từ trái sang phải và từ trên xuống dưới. Mỗi bước dịch 1 đơn vị theo stride $s = 1$. Thực hiện lại phép tích chập tương tự ở các vị trí tiếp theo.

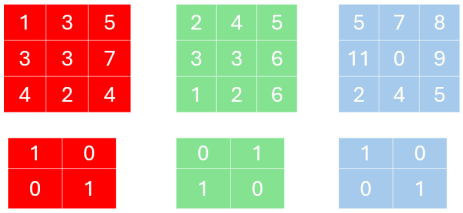
Kết luận: Phép tích chập cho phép mạng học được các đặc trưng quan trọng bằng cách nhân từng vùng ảnh với bộ lọc và tổng hợp lại.



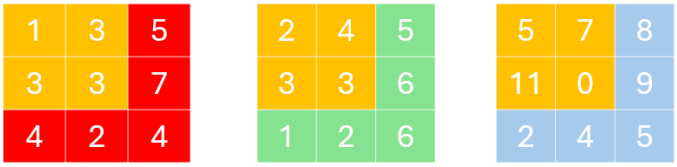
(a) Bản đồ đặc trưng có 3 kênh màu



(b) Bộ lọc có 3 kênh màu tương ứng



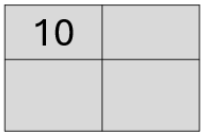
(c) Bộ lọc tương ứng với từng kênh màu



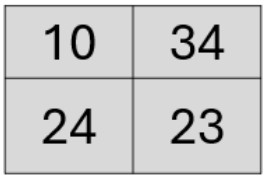
$$c_{1,1,1} = a_{1,1,1} \cdot f_{1,1,1} + a_{1,2,1} \cdot f_{1,2,1} + a_{2,1,1} \cdot f_{2,1,1} + a_{2,2,1} \cdot f_{2,2,1}$$

$$o_{1,1} = c_{1,1,1} + c_{1,1,2} + c_{1,1,3}$$

(d) Công thức tính giá trị đầu ra ở vị trí (1,1)



(e) Đáp án đầu ra tại vị trí (1,1)



(f) Kết quả đầu ra đầy đủ

Hình 2.1. Minh họa các bước trong phép tích chập từ đầu vào đến đầu ra cuối cùng

2.1.2 Tổng hợp cực đại (Max pooling)

Tổng hợp cực đại là một phép toán tổng hợp dùng để chọn giá trị lớn nhất của từng vùng nhỏ trên bản đồ đặc trưng (feature map). Kết quả là một bản đồ đặc trưng mới có kích thước giảm đi, chỉ giữ lại những đặc trưng nổi bật nhất. Nhờ đó, bản đồ mới không chỉ gọn hơn mà còn tập trung vào các thông tin quan trọng nhất từ bản đồ đặc trưng ban đầu.

Mục đích của phép tổng hợp cực đại:

- Giảm kích thước không gian của bản đồ đặc trưng.
- Giảm số lượng tham số và chi phí tính toán.
- Hạn chế hiện tượng khớp quá mức (overfitting).
- Tăng tính bất biến đối với dịch chuyển nhỏ (tăng invariance với dịch chuyển ảnh).

Thuật toán tổng hợp cực đại:
Giả sử:

- Đầu vào (input): một bản đồ đặc trưng 2 chiều có kích thước $h \times w$
- Bộ lọc (filter): kích thước $k \times k$
- Số bước nhảy (stride): s
- Vùng đệm (padding): p (thường bằng 0)

Các bước thực hiện:

1. Di chuyển bộ lọc $k \times k$ qua bản đồ đặc trưng với bước nhảy s.
2. Tại mỗi vị trí, lấy giá trị lớn nhất trong vùng $k \times k$.
3. Ghi giá trị đó vào bản đồ đặc trưng mới.
4. Lặp lại cho đến khi quét hết vùng ảnh.

Kích thước đầu ra được tính bằng công thức sau:

$$H_{\text{out}} = \left\lfloor \frac{H - k + 2p}{s} \right\rfloor + 1$$

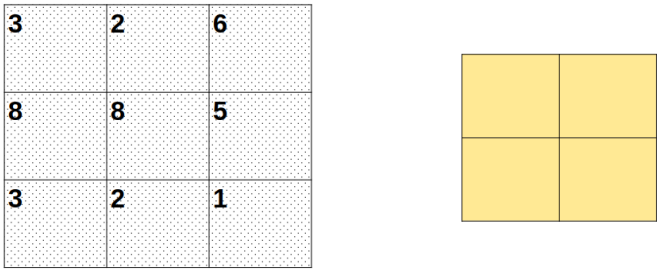
$$W_{\text{out}} = \left\lfloor \frac{W - k + 2p}{s} \right\rfloor + 1$$

Trong đó:

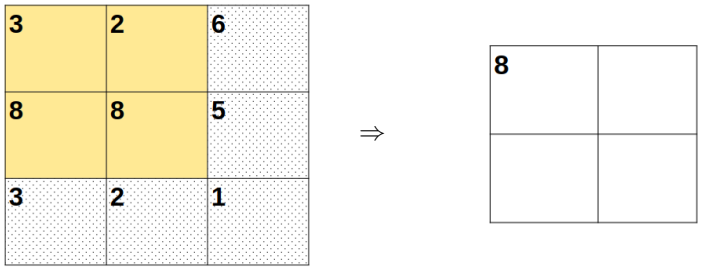
- H, W là chiều cao và chiều rộng của bản đồ đặc trưng đầu vào
- k là kích thước kernel (bộ lọc)
- p là padding
- s là stride
- H_{out}, W_{out} là chiều cao và chiều rộng của bản đồ đầu ra

Giả sử bước nhảy $s = 1$, và vùng đệm $p = 0$. Hình minh họa sau sẽ trình bày quá trình tạo ra bản đồ đặc trưng mới bằng phép pooling.

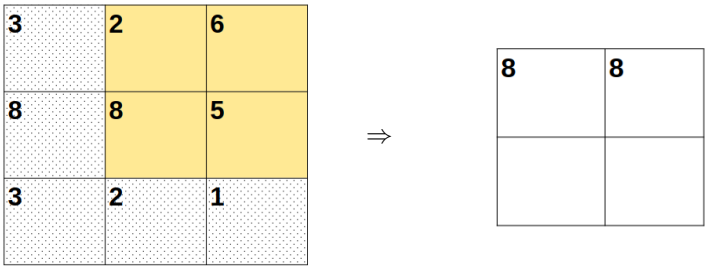
Bản đồ đặc trưng kích thước 3×3 và bộ tổng hợp cực đại 2×2



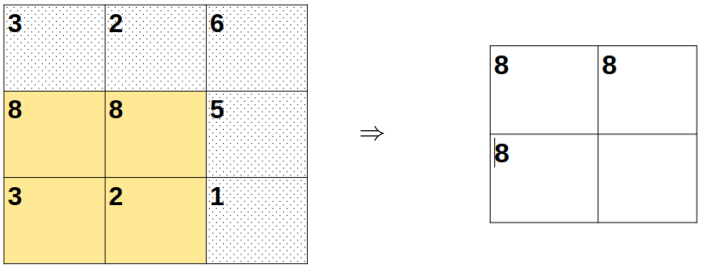
Bước đầu tiên: Quét vùng đầu tiên và tính ra giá trị



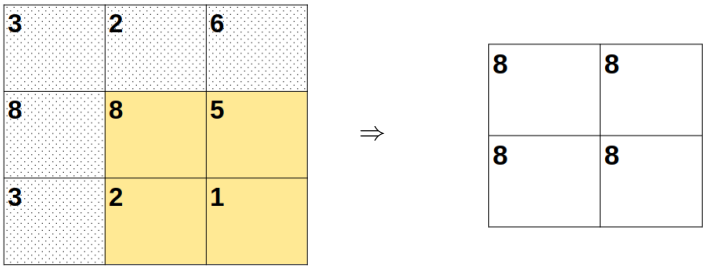
Tiếp theo: Dịch 1 đơn vị sang phải



Khi quét hết hàng: Dịch 1 đơn vị xuống dưới



Cuối cùng: Dịch sang phải, tính giá trị cuối cùng



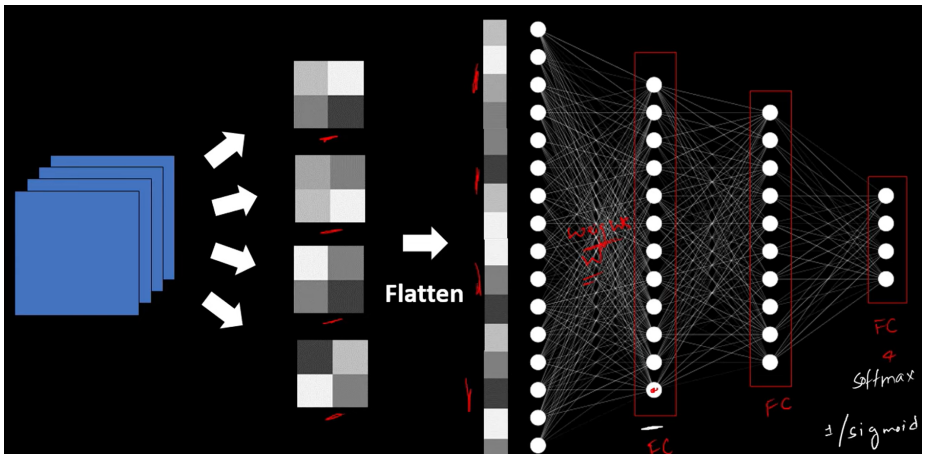
2.1.3 Lớp Kết nối đầy đủ (Fully Connected Layer)

Lớp kết nối đầy đủ hay còn gọi là lớp dày đặc (*dense layer*) trong mạng nơ-ron là tầng đơn giản nhất, nhưng cũng là quan trọng nhất. Nó giống như là phần cuối, dùng để "tổng hợp" các đặc trưng quan trọng đã trích xuất từ trước, để đưa ra dự đoán cuối cùng.

Ví dụ như trong kiến trúc mạng tích chập, đây là phần cuối để "tổng hợp" toàn bộ thông tin rút ra từ các lớp trước (tích chập, tổng hợp, …) rồi đưa ra dự đoán cuối cùng.

Như ta đã thấy trong hình, bản đồ đặc trưng 3D có kích thước $h \times w \times c$, gồm có c bản đồ đặc trưng 2D kích thước $h \times w$. Chúng được làm phẳng (flatten), kết nối lại thành một vector cột. Mỗi phần tử trong vector này là một nơ-ron của lớp đầu tiên.

Tất cả các nơ-ron đó được kết nối đầy đủ với từng nơ-ron trong các lớp kế tiếp — được gọi là **lớp kết nối đầy đủ** (*fully connected layer*). Mỗi nơ-ron này được tính thông qua hàm kích hoạt như ReLU. Cứ thế cho đến lớp kết nối cuối cùng, gồm n nơ-ron tương ứng với n lớp cần phân loại. Vì có nhiều hơn 2 lớp nên ta sẽ sử dụng hàm **Softmax**.



Hình 2.2. Hình mô tả công việc của các lớp kết nối đầy đủ

2.1.4 Hàm Softmax:

Hàm Softmax được sử dụng để dự đoán đầu vào thuộc lớp nào. Kết quả là xác suất của đầu vào thuộc một trong n lớp cần dự đoán. Lớp có xác suất cao nhất sẽ được chọn làm kết quả cuối cùng.

$$a_i = \frac{\exp(z_i)}{\sum_{j=1}^{C} \exp(z_j)}, \quad \forall i = 1, 2, \ldots, C$$

$$z_i = \mathbf{w}_i^T \mathbf{x}$$

Hàm mất mát – Categorical Cross-Entropy (CCE):

CCE đo độ lệch giữa phân phối xác suất thật và phân phối xác suất dự đoán.

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \cdot \log(\hat{y}_i)$$

- y_i: Nhãn thật (theo định dạng one-hot vector)
- $\hat{y}_i$: Xác suất dự đoán từ hàm softmax
- C: Số lớp

2.1.5 Hàm ReLU:

Hàm kích hoạt ReLU (Rectified Linear Unit) là một trong những hàm được sử dụng phổ biến nhất nhờ sự đơn giản và hiệu quả.

$$g(x) = \max(0, x)$$

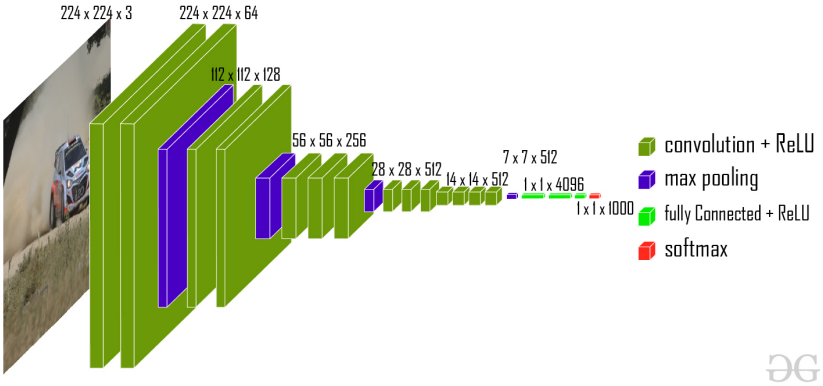
2.2 VGG16

2.2.1 Giới thiệu chung:

VGG16 là kiến trúc được đề xuất bởi Simonyan và Zisserman trong bài báo "Very Deep Convolutional Network" [3]. Mục tiêu chính là khám phá tác động của độ sâu đến khả năng biểu diễn của mạng trong bài toán nhận dạng hình ảnh quy mô lớn (*ImageNet*).

VGG16 được xây dựng dựa trên các ý tưởng sau:

- **Xếp chồng các bộ lọc nhỏ:** Thay vì sử dụng một bộ lọc lớn (7×7), VGG16 sử dụng nhiều lớp tích chập (convolution) liên tiếp với bộ lọc 3×3. Cách làm này giúp đạt được trường tiếp nhận (*receptive field*) tương đương với bộ lọc lớn, nhưng có nhiều ưu điểm:
  - Tăng tính phi tuyến nhờ có nhiều lớp ReLU xen kẽ.
  - Giảm số lượng tham số học do kích thước bộ lọc nhỏ hơn.
- **Độ sâu của mạng:** Với tổng cộng 16 lớp có trọng số (13 lớp convolution và 3 lớp fully-connected), VGG16 cho thấy rằng việc tăng độ sâu giúp mô hình học được các biểu diễn phức tạp và khả năng phân biệt tốt hơn.



Hình 2.3. Minh họa luồng xử lý đầu vào qua các tầng của VGG16 với kích thước tương ứng

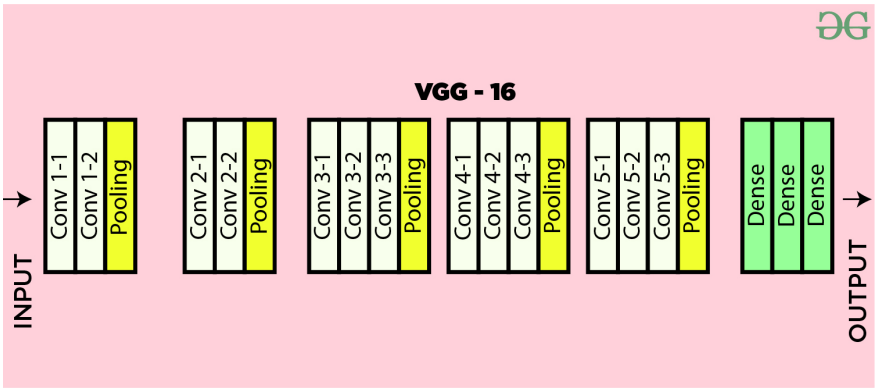
2.2.2 Kiến trúc chi tiết:

Mỗi lớp convolution trong VGG16 sử dụng các bộ lọc kích thước 3×3, với bước dịch (*stride*) bằng 1 và padding phù hợp để bảo toàn kích thước không gian của đầu vào.

Nhờ cách thiết kế này, các lớp kế tiếp có thể "nhìn" vào các vùng nhỏ trong ảnh, nhưng thông qua tích lũy qua nhiều lớp, mạng có thể học các đặc trưng ở cấp độ rộng hơn.

Sau các lớp convolution, VGG16 sử dụng 3 lớp fully-connected:

- Hai lớp đầu có 4096 đơn vị dùng để biến đổi và trích xuất các đặc trưng tổng hợp.
- Lớp cuối cùng là lớp *softmax* với 1000 đơn vị đầu ra tương ứng với 1000 lớp trong bộ dữ liệu ImageNet [2][3].



Hình 2.4. Kiến trúc tầng lớp của VGG16: các khối convolution, pooling và fully-connected

Tất cả các lớp ẩn đều sử dụng hàm kích hoạt ReLU, giúp tăng khả năng biểu diễn phi tuyến và học các mối quan hệ phức tạp trong dữ liệu.

Chương 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ MÔ HÌNH

3.1 Dữ liệu và phương pháp thu thập

Dữ liệu được thu thập để xây dựng mô hình nhận dạng tiền Việt Nam thông qua việc quay video bằng webcam (sử dụng `cv2.VideoCapture(0)`). Mỗi frame video được trích xuất thành ảnh để tạo tập dữ liệu. Quá trình này được thực hiện như sau:

- **Quy trình thu thập:** Webcam bắt đầu quay và hiển thị khung hình qua `cv2.imshow()`. Sau 60 ảnh đầu tiên (để tránh ghi lại các khung hình chưa sẵn sàng), mỗi ảnh được chỉnh sửa kích thước với tỷ lệ 0.3 lần kích thước gốc bằng `cv2.resize()` và lưu vào thư mục tương ứng với nhãn (ví dụ: `data/500000` cho mệnh giá 500.000 VNĐ).
- **Nhãn dữ liệu:** Nhãn được định nghĩa thủ công (ví dụ: `label = "500000"`) và `"00000"` đại diện cho trường hợp không cầm tiền. Ảnh được lưu dưới định dạng `.png` với tên file là số thứ tự ảnh (ví dụ: `"60.png"`).
- **Tổ chức dữ liệu:** Các thư mục được tạo tự động nếu chưa tồn tại bằng `os.mkdir()`, đảm bảo dữ liệu được phân loại theo nhãn.

Tập dữ liệu sau đó được chia thành tập huấn luyện và tập kiểm tra với tỷ lệ 80:20 thông qua hàm `train_test_split()` .
Dưới đây là hình ảnh mẫu của từng nhãn:

Hình 3.1. Mẫu hình ảnh của từng nhãn

3.2 Tiền xử lý và tăng cường dữ liệu

Dữ liệu thô từ các thư mục được tiền xử lý để chuẩn bị cho việc huấn luyện mô hình:

- **Resize ảnh:** Ảnh được đọc bằng `cv2.imread()` và resize về kích thước (128, 128) bằng `cv2.resize()` trong hàm `save_data()`.
- **Mã hóa nhãn:** Nhãn được chuyển thành dạng one-hot encoding bằng `LabelBinarizer` từ `sklearn`, phù hợp với bài toán phân loại đa lớp.
- **Lưu trữ dữ liệu:** Dữ liệu ảnh (pixels) và nhãn (labels) được lưu vào file `pix.data` bằng `pickle.dump()`, cho phép tái sử dụng thông qua hàm `load_data()`.

Để tăng khả năng tổng quát hóa của mô hình, dữ liệu được tăng cường bằng `ImageDataGenerator` với các tham số:

- Xoay ngẫu nhiên tối đa 20 độ (`rotation_range=20`).
- Phóng to/thu nhỏ trong khoảng 10% (`zoom_range=0.1`).
- Dịch chuyển ngang và dọc tối đa 10% (`width_shift_range=0.1, height_shift_range=0.1`).
- Lật ngang (`horizontal_flip=True`).
- Điều chỉnh độ sáng từ 0.2 đến 1.5 (`brightness_range=[0.2,1.5]`).
- Chuẩn hóa giá trị pixel về [0-1] (`rescale=1./255`).

Mô hình này được áp dụng trong quá trình huấn luyện thông qua `aug.flow()`.

(a) Ảnh gốc

(b) Xoay ảnh gốc 10 độ

(c) Lật ngang ảnh gốc

(d) Dịch chuyển ngang và dọc 5%

Hình 3.2. Các phiên bản tăng cường của ảnh 5.000 VNĐ

3.3 Huấn luyện mô hình

Mô hình được xây dựng dựa trên VGG16 với transfer learning trong hàm `get_model()`:

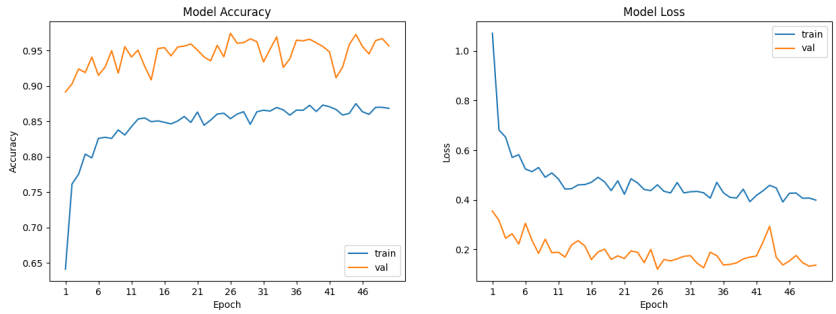
- **Kiến trúc:**
  - Sử dụng VGG16 (`weights='imagenet', include_top=False`) làm backbone.
  - Đóng băng tất cả các lớp của VGG16 (`layer.trainable = False`).
  - Thêm các lớp: `Flatten`, hai lớp `Dense` (4096 nơ-ron, ReLU) với `Dropout(0.5)`, và lớp đầu ra `Dense(9, softmax)` cho 9 nhãn.
- **Huấn luyện:**
  - Hàm mất mát: `categorical_crossentropy`.
  - Tối ưu hóa: `adam`.
  - Batch size: 16.
  - Epoch: 50.
  - Lưu trọng số tốt nhất vào file `.hdf5` dựa trên `val_loss` qua `ModelCheckpoint`.

3.4 Đánh giá mô hình và kết quả phân tích

Hiệu suất của mô hình được đánh giá dựa trên hai chỉ số chính:

- **Accuracy (Độ chính xác):** Đo lường mức độ chính xác của mô hình trên cả tập huấn luyện và kiểm tra qua từng epoch.
- **Loss (Hàm mất mát):** Đánh giá mức độ sai lệch giữa dự đoán và giá trị thực tế, phản ánh hiệu quả học của mô hình.

Hàm `plot_model_history()` được sử dụng để trực quan hóa quá trình huấn luyện, bằng cách vẽ biểu đồ thể hiện sự thay đổi của độ chính xác và hàm mất theo từng epoch cho cả tập huấn luyện và tập kiểm tra.



Hình 3.3. Biểu đồ Accuracy và Loss theo Epoch

Phân tích biểu đồ:

- **Accuracy:**
  - Độ chính xác trên tập huấn luyện tăng dần từ khoảng 65% đến khoảng 85% sau 50 epoch.
  - Trong khi đó, độ chính xác trên tập kiểm tra ban đầu ở mức cao (khoảng 90%) và duy trì ổn định, dao động nhẹ quanh mức 90-95
  - Điều này cho thấy mô hình học hiệu quả và có khả năng tổng quát tốt, không có dấu hiệu rõ ràng của overfitting.
- **Loss:**
  - Hàm mất mát trên tập huấn luyện giảm đều từ hơn 1.0 xuống khoảng 0.4, minh chứng cho việc mô hình cải thiện dần qua từng epoch.
  - Ngược lại, loss trên tập kiểm tra giảm mạnh xuống mức thấp (dao động từ 0.1 đến 0.3) và ổn định trong suốt quá trình huấn luyện.
  - Sự khác biệt giữa loss của hai tập là nhỏ, cho thấy mô hình không bị hiện tượng underfitting hay overfitting đáng kể.

Biểu đồ trên cho thấy mô hình đạt hiệu quả học tập tốt, với độ chính xác cao trên cả hai tập dữ liệu và loss giảm đều đặn. Tuy nhiên, để tối ưu hơn nữa, có thể thử nghiệm thêm các điều chỉnh như:

- Tinh chỉnh các siêu tham số (learning rate, batch size, ...)
- Áp dụng kỹ thuật regularization nhẹ để tránh overfitting trong các lần huấn luyện dài hơn.
- Kiểm tra bổ sung với các bộ dữ liệu ngoài để xác thực tính ổn định của mô hình.

Lưu ý:

- Để huấn luyện được mô hình cho độ chính xác tốt như này. Tôi đã phải thu thập tập dữ liệu để huấn luyện nhiều lần. Và qua các lần thu thập đó, tôi rút ra được kết luận sau:
  - Khi thu thập dữ liệu, đối tượng nhận dạng trong ảnh nên xe dịch ít đi. Để ta có thể thu được lượng lớn dữ liệu, mà đối tượng gần như ở cùng 1 vị trí, từ đó có thể học được toàn diện đặc trưng mà ta cần, rồi mới di chuyển đến vị trí khác.
  - Việc thu thập dữ liệu tính toán điều kiện trước để phù hợp môi trường thực rất khó khăn. Nên ta cần phải có một phương pháp nào đó để thu thập được dữ liệu cho mô hình học được tổng quát hơn.
- Để huấn luyện và tinh chỉnh được các mô hình học sâu này là một điều khó khăn:
  - Trong phần huấn luyện mô hình này, tôi có ý định sẽ huấn luyện nhiều mô hình bằng cách tỉnh chỉnh việc đóng băng các tham số của mô hình VGG16. Nhưng bởi vì GPU (3050 Ti laptop) không thể sử lý được số lượng tham số lớn như vậy, nên chỉ có thể đóng băng toàn bộ tham số, ngoại trừ lớp dự đoán cuối cùng.
  - Cũng như giới hạn của GPU như trên, nên vấn đề thay đổi tham số `epoch` hay `batch_size` cũng bị giới hạn. Điển hình là tôi chỉ cho mô hình học ở 50 `epoch` và 16 `batch_size`.

KẾT LUẬN

Trong bối cảnh chuyển đổi số và ứng dụng trí tuệ nhân tạo ngày càng mở rộng, việc tự động hóa quy trình nhận dạng tiền tệ mang lại nhiều lợi ích thiết thực cho đời sống và kinh tế. Đề tài "Nhận dạng tiền Việt Nam bằng mô hình VGG16 qua camera" đã tập trung nghiên cứu và triển khai một hệ thống ứng dụng học sâu để giải quyết bài toán nhận dạng mệnh giá tiền Việt Nam.

Trong quá trình thực hiện, mô hình mạng nơ-ron tích chập VGG16 – một trong những kiến trúc CNN mạnh mẽ và phổ biến – đã được sử dụng làm nền tảng chính. Hệ thống đã trải qua các giai đoạn tiền xử lý dữ liệu, huấn luyện mô hình, đánh giá và kiểm thử. Kết quả huấn luyện cho thấy độ chính xác của mô hình đạt mức cao (khoảng 95%), chứng tỏ khả năng học và phân loại tốt các đặc trưng của từng mệnh giá.

Bên cạnh những kết quả đạt được, hệ thống vẫn còn một số hạn chế như: mô hình có thể bị ảnh hưởng khi ánh sáng thay đổi mạnh, ảnh bị mờ, hoặc tiền bị che khuất một phần. Ngoài ra, việc nhận dạng các mệnh giá có thiết kế màu sắc và bố cục tương đồng đôi khi còn gây nhầm lẫn.

Định hướng phát triển:

- Tăng kích thước tập dữ liệu huấn luyện với đa dạng điều kiện ánh sáng, góc chụp, độ mờ.
- Tích hợp thêm các kỹ thuật xử lý ảnh như histogram equalization, edge enhancement để tăng chất lượng đầu vào.
- Thử nghiệm với các kiến trúc tiên tiến hơn như ResNet, EfficientNet hoặc MobileNet để so sánh hiệu năng.
- Xây dựng giao diện ứng dụng thực tế (mobile/web) nhằm áp dụng vào ATM, máy đếm tiền, hoặc hệ thống bán hàng tự động.
- Thử nghiệm ba chiến lược huấn luyện với kỹ thuật học chuyển giao: (1) không đóng băng lớp nào, (2) đóng băng một nửa số lớp đầu, và (3) đóng băng toàn bộ các lớp, chỉ huấn luyện lớp phân loại cuối cùng.

Tổng thể, đề tài đã đạt được mục tiêu đề ra, góp phần khẳng định hiệu quả của mô hình học sâu – đặc biệt là VGG16 – trong bài toán nhận dạng tiền Việt Nam. Đây sẽ là tiền đề quan trọng cho các nghiên cứu và ứng dụng thực tế tiếp theo trong lĩnh vực thị giác máy tính.

TÀI LIỆU THAM KHẢO

Tiếng Việt:
Tiếng Anh:

[1] César G. Pachón, Dora M. Ballesteros, Diego Renza (2021). Fake Banknote Recognition Using Deep Learning. *Applied Sciences*.
https://doi.org/10.3390/app11031281
[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, Li Fei-Fei (2009). ImageNet: A Large-Scale Hierarchical Image Database. *IEEE Conference on Computer Vision and Pattern Recognition - CVPR*.
https://image-net.org/static_files/papers/imagenet_cvpr09.pdf
[3] Karen Simonyan, Andrew Zisserman (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv*.
https://doi.org/10.48550/arXiv.1409.1556
[4] Muhammad Sarfraz (2015). An Intelligent Paper Currency Recognition System. *Procedia Computer Science*.
https://doi.org/10.1016/j.procs.2015.09.128
[5] P. Divya Jenifar, V. Harinitha, M.D. Harshini, A. James Christiya (2023). Currency Detection and Recognition System Based on Deep Learning. *International Journal of Scientific Research in Engineering and Management (IJSREM)*.
https://scispace.com/pdf/currency-detection-and-recognition-system-based-on-deep-6q2ea8kk.pdf

PHỤ LỤC

Toàn bộ mã nguồn và hình ảnh phục vụ quá trình huấn luyện và kiểm thử mô hình được lưu trữ tại:
`https://github.com/htrongnghia/Currency_Recognition_VGG16`

TRƯỜNG ĐẠI HỌC KHOA HỌC
KHOA .....

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

PHIẾU ĐÁNH GIÁ TIỂU LUẬN

Học kỳ ..... **Năm học** .....–.....

Cán bộ chấm thi 1	**Cán bộ chấm thi 2**
Nhận xét:	**Nhận xét:** . . . . .
. . . . .	. . . . .
. . . . .	. . . . .
. . . . .	. . . . .
. . . . .	. . . . .
. . . . .	. . . . .
. . . . .	. . . . .
. . . . .	. . . . .
. . . . .	. . . . .
. . . . .	. . . . .
. . . . .	. . . . .
. . . . .	. . . . .
Điểm đánh giá của CBChT1:	**Điểm đánh giá của CBChT1:**
Bằng số:	Bằng số: . . . . .
Bằng chữ:	Bằng chữ: . . . . .

Điểm kết luận: Bằng số: . . . . . Bằng chữ: . . . . . .

Thừa Thiên Huế, ngày . . . tháng . . . năm 20 . . .

CBChT1

(Ký và ghi rõ họ tên)

CBChT2

(Ký và ghi rõ họ tên)