

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH

KHOÁ LUẬN TỐT NGHIỆP

TÊN ĐỀ TÀI

KẾT HỢP RÚT TRÍCH ĐẶC TRƯNG CỤC BỘ
VÀ HỌC SÂU ĐỂ GIẢI QUYẾT BÀI TOÁN
PHÂN LOẠI VẬT LIỆU DỰA TRÊN HÌNH ẢNH

Khoa: Công nghệ thông tin

Sinh viên thực hiện: Phan Văn Hoài Đức

Người hướng dẫn: TS. Trương Hoàng Vinh

TP. Hồ Chí Minh, tháng 3 năm 2019

LỜI CẢM ƠN

Em xin gửi đến quý thầy cô ở Khoa Công nghệ thông tin, Trường Đại Học Mở Thành Phố Hồ Chí Minh đã cùng với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Em xin gửi lời cảm ơn chân thành đến thầy Trương Hoàng Vinh, thầy đã tận tâm hướng dẫn chúng em trong suốt quá trình nghiên cứu về xử lý ảnh thông qua những buổi nói chuyện, thảo luận. Nếu không có những lời hướng dẫn, dạy bảo của thầy thì em nghĩ bài thu hoạch này của em rất khó có thể hoàn thiện được.

Sau cùng, em xin kính chúc quý thầy cô trong khoa Công Nghệ thông tin và thầy hiệu trưởng thật dồi dào sức khỏe để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Mục lục

Danh sách thuật ngữ tiếng Anh	1
Danh sách ký hiệu	2
Danh sách từ viết tắt	3
Danh sách hình vẽ	4
Danh sách bảng	6
Tóm tắt khoá luận	7
Tổng quan về khoá luận	8
1 Cơ sở lý thuyết	10
1.1 Một số khái niệm cơ bản	10
1.1.1 Ảnh đa mức xám (Grayscale Image)	10
1.1.2 Ảnh màu	10
1.1.3 Định nghĩa kết cấu (Texture)	11
1.1.4 Định nghĩa phân loại kết cấu (Texture Classification)	11
1.2 Local Binary Pattern (LBP)	11
1.2.1 Định nghĩa	11
1.2.2 Ưu điểm và nhược điểm	13
1.2.3 Một số biến thể LBP	13
1.3 Học máy (Machine Learning)	19
1.3.1 Định nghĩa	19
1.3.2 Một số thuật toán	19
2 Mạng neuron nhân tạo (Artificial Neural Networks)	24
2.1 Định nghĩa	24
2.2 Một số thành phần quan trọng	25
2.2.1 Hàm Sigmoid	25
2.2.2 Hàm Rectified Linear Units (ReLU)	26

2.2.3	Hàm Softmax	26
2.2.4	Hàm mất mát (Loss Function)	27
2.2.5	Thuật toán lan truyền ngược (Backpropagation Algorithm)	28
2.3	Học sâu (Deep Learning)	29
2.3.1	Định nghĩa	29
2.3.2	Lịch sử hình thành, phát triển	29
2.3.3	Mạng neuron tích chập (Convolutional Neural Network - CNN)	30
3	Phương pháp đề xuất	37
3.1	Tiền xử lý dữ liệu	37
3.1.1	Đặc trưng Neighbor-Center Difference Image (NCDI)	37
3.1.2	Đặc trưng NCDI histogram và LBP NCDI	38
3.1.3	Đặc trưng Enhanced NCDI (ENCDI)	38
3.1.4	Kỹ thuật Multi-crop	39
3.2	Mô hình đề xuất	40
4	Quá trình thực nghiệm	41
4.1	Các tập dữ liệu	41
4.1.1	New-BarkTex	41
4.1.2	Outex-TC00013	42
4.1.3	USPTex	44
4.1.4	STex	45
4.2	Thiết lập cho các thí nghiệm	46
4.2.1	Thiết lập cho thí nghiệm đánh giá đặc trưng NCDI histogram và LBP NCDI	46
4.2.2	Thiết lập cho thí nghiệm đánh giá mô hình học sâu kết hợp đặc trưng cục bộ	46
4.3	Kết quả	47
4.3.1	Kết quả đánh giá đặc trưng NCDI histogram và LBP NCDI	47
4.3.2	Kết quả đánh giá mô hình học sâu kết hợp đặc trưng cục bộ	48
5	Kết luận	51
Tài liệu tham khảo		67

Danh sách thuật ngữ tiếng Anh

Activation Function	Hàm kích hoạt
Artificial Neural Networks	Mạng neuron nhân tạo
Axon	Sợi trục
Backpropagation Algorithm	Thuật toán lan truyền ngược
Convolutional Neural Network	Mạng neuron tích chập
Cross-Entropy Loss	Hàm đánh giá độ măt măt Cross-Entropy
Deep Learning	Học sâu
Dendrite	Sợi nhánh
Fully-connected	Kết nối đầy đủ
Gradient	Độ dốc
Grayscale Image	Ảnh đa mức xám
Hàm Softmax	Hàm phân phối xác suất Softmax
Histogram	Biểu đồ thông kê tần xuất
ImageNet Large Scale Visual Recognition Challenge	Cuộc thi phân loại ảnh quy mô lớn ImageNet
Local Binary Pattern	Mẫu nhị phân địa phương
Local Ternary Pattern	Mẫu tam phân địa phương
Loss Function	Hàm măt măt
Machine Learning	Học máy
Neighbor-Center Difference Image	Hình ảnh khác biệt giữa điểm ảnh trung tâm so với điểm ảnh lân cận
One-versus-all	Một so với tất cả
One-versus-one	Một so với một
Pixel	Điểm ảnh
Rectified Linear Units	Tinh chỉnh đơn vị tuyến tính
Sliding Windows	Cửa sổ trượt
Stride	Khoảng cách dịch chuyển
Supervised Learning	Học có giám sát
Support Vector Machine	Máy vector hỗ trợ
Texture	Kết cấu
Texture Classification	Bài toán phân loại kết cấu
Transition	Điểm chuyển tiếp
Unsupervised Learning	Học không giám sát
Unsupervised Pretraining	Tiền huấn luyện không giám sát

Danh sách ký hiệu

A_i	Giá trị thứ i của dãy kết quả phân ngưỡng dùng để tính giá trị LTP
A_l	Dãy kết quả phân ngưỡng dùng để tính giá trị LTP phần dưới
A_u	Dãy kết quả phân ngưỡng dùng để tính giá trị LTP phần trên
b	Tham số bias
c	Giá trị phân ngưỡng
c_m	Giá trị trung bình của tất cả điểm ảnh trong một ảnh
D	Vị trí điểm mẫu có độ chênh lệch lớn nhất so với điểm ảnh trung tâm
d	Số kênh của dữ liệu đầu vào
d_i	Giá trị chênh lệch giữa điểm mẫu thứ i so với điểm ảnh trung tâm
d_o	Số kênh của dữ liệu đầu ra
g_c	Giá trị của điểm mẫu trung tâm
g_i	Giá trị của điểm mẫu thứ i
H	Chiều cao của ma trận đầu vào
H_o	Chiều cao của ma trận đầu ra
L	Giá trị mất mát
l	Kích thước của ma trận tích chập
l_s	Kích thước cửa sổ trượt
N	Số lượng ma trận tích chập
$\text{NCDI}(x, y)_i$	Giá trị NCDI tại vị trí (x, y) của kênh thứ i
P	Độ dày của khung Zero-padding
p	Phần trăm loại bỏ của lớp Dropout
R	Bán kính để lấy các điểm mẫu
S	Khoảng cách dịch chuyển của ma trận tích chập
S_p	Khoảng cách dịch chuyển của cửa sổ trượt
S_t	Dãy kết quả sau khi phân ngưỡng
W	Chiều rộng của ma trận đầu vào
W_o	Chiều rộng của ma trận đầu ra
w	Tham số weight
$\frac{\partial f}{\partial x}$	Giá trị đạo hàm riêng của f theo x

Danh sách từ viết tắt

ANN	Artificial Neural Networks
CLBP	Complete LBP
CNN	Convolutional Neural Network
Conv	Convolutional Layer
DBN	Deep Belief Network
ENCDI	Đặc trưng Enhanced Neighbor-Center Difference Image
FC	Fully-connected
GPU	Graphics Processing Unit - Bộ xử lý đồ họa
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
KNN	K-Nearest Neighbors
LBP	Local Binary Pattern
LTP	Local Ternary Pattern
NCDI	Đặc trưng Neighbor-Center Difference Image
Pool	Pooling Layer
ReLU	Rectified Linear Units
RGB	Ảnh màu RGB
RLBP	Rotated LBP
SVM	Support Vector Machine

Danh sách hình vẽ

1.1	Hình minh họa ảnh xám và ảnh màu	10
1.2	Một số hình trong bộ dữ liệu Outex-TC00013	11
1.3	Hình minh họa quá trình trích xuất giá trị LBP _{8,1}	12
1.4	Hình các LBP Uniform	14
1.5	Hình minh họa quá trình trích xuất hai giá trị LTP _{8,1}	15
1.6	Hình minh họa quá trình trích xuất hai giá trị RLBP _{8,1}	17
1.7	Hình minh họa thuật toán KNN	20
1.8	Minh họa trạng thái khởi tạo của thuật toán <i>K</i> -Means Clustering . .	21
1.9	Cập nhật các điểm trung tâm trong thuật toán <i>K</i> -Means Clustering .	21
1.10	Minh họa phân loại hai lớp bằng thuật toán SVM	22
1.11	Minh họa SVM phân loại nhiều lớp với chiến thuật Một so với tất cả cơ bản	23
1.12	Minh họa SVM phân loại nhiều lớp với chiến thuật Một so với tất cả	23
2.1	Hình minh họa một neuron sinh học (nguồn http://cs231n.github.io/)	24
2.2	Hình minh họa một mạng neuron nhân tạo gồm 2 lớp FC	25
2.3	Đồ thị hàm Sigmoid	26
2.4	Đồ thị hàm ReLU	26
2.5	Phân phối xác suất bằng hàm Softmax	27
2.6	Đồ thị hàm Log Loss	28
2.7	Minh họa cách tính giá trị mất mát từ kết quả phân phối xác suất .	28
2.8	Minh họa mô hình CNN	31
2.9	Minh họa ý nghĩa của các lớp Convolutional trong mô hình CNN (Nguồn từ [1])	32
2.10	Minh họa cách hoạt động của lớp Convolutional (1)	33
2.11	Minh họa cách hoạt động của lớp Convolutional (2)	33
2.12	Minh họa cách hoạt động của lớp Convolutional (3)	34
2.13	Minh họa cách hoạt động của lớp Max Pooling	35
2.14	Minh họa cách hoạt động của lớp Global Average Pooling	35
2.15	Minh họa mô hình có sử dụng và không sử dụng Dropout	36

3.1	Minh họa quá trình trích xuất đặc trưng NCDI 8 kênh	37
3.2	Minh họa quá trình trích xuất đặc trưng ENCDI	38
3.3	Minh họa cách cắt 5 hình nhỏ từ một hình gốc	39
3.4	Hình ảnh minh họa mô hình được đề xuất.	40
4.1	Cách lấy ảnh của bộ dữ liệu New BarkTex	42
4.2	Một số ảnh mẫu trong bộ dữ liệu New BarkTex	42
4.3	Một số ảnh mẫu trong bộ dữ liệu Outex-TC00013	43
4.4	Một số ảnh mẫu trong bộ dữ liệu USPTex	44
4.5	Một số ảnh mẫu trong bộ dữ liệu STex	45

Danh sách bảng

4.1	Bảng tóm tắt thông tin của bốn bộ dữ liệu được sử dụng trong quá trình thực nghiệm.	41
4.2	Bảng thống kê độ chính xác (tính bằng %) của LBP và các phương pháp đề xuất trên bốn bộ dữ liệu gồm New-BarkTex, Outex-TC00013, USPTex và STex.	47
4.3	Bảng thống kê độ chính xác (tính bằng %) của phương pháp đề xuất và một số phương pháp khác trên bốn bộ dữ liệu New BarkTex, Outex-TC-00013, USPTex, và STex.	48
4.4	Bảng thống kê các thuật toán có độ chính xác (tính bằng %) cao nhất hiện tại và kết quả của mô hình đề xuất trên bốn bộ dữ liệu gồm New-BarkTex, Outex-TC00013, USPTex và STex.	49

TÓM TẮT KHOÁ LUẬN

Bài toán phân tích texture có rất nhiều ứng dụng thực tế như phân loại vật liệu, nhận diện khuôn mặt, phân vùng vật thể trong ảnh. Trong khoá luận này, nhóm đã nghiên cứu và đề xuất được hai loại đặc trưng mới và một hướng giải quyết mới cho bài toán phân loại texture. Trong thời gian gần đây, hướng giải quyết dùng các mô hình neuron tích chập đã đạt được những kết quả tốt hơn trong bài toán phân tích texture và những vấn đề khác của lĩnh vực thị giác máy tính. Mô hình neuron tích chập thông thường dùng ảnh RGB làm dữ liệu đầu vào. Tuy nhiên, việc sử dụng một số loại ảnh đặc trưng làm dữ liệu phụ trợ cho mô hình neuron tích chập đã nâng cao độ chính xác cho mô hình. Từ đó, nhóm đã nghiên cứu và đề xuất một phương pháp để nâng cao độ chính xác cho các mô hình neuron tích chập bằng cách sử dụng ảnh đặc trưng làm dữ liệu phụ trợ. Qua quá trình thực nghiệm cho thấy phương pháp được đề xuất đạt được kết quả tốt hơn trên bốn bộ dữ liệu texture màu.

TỔNG QUAN VỀ KHOÁ LUẬN

Xử lý ảnh là một phân ngành khoa học mới rất phát triển trong những năm gần đây. Sự phát triển của xử lý ảnh đã và đang mang lại rất nhiều lợi ích cho cuộc sống hàng ngày của chúng ta. Một trong các lĩnh vực quan trọng nhất trong xử lý ảnh là phân loại kết cấu (Texture Classification), đây là một lĩnh vực nghiên cứu có thể áp dụng cho nhiều ứng dụng thực tế như kiểm tra bề mặt vật liệu trong công nghiệp, phát hiện bệnh trong y học, phân loại các loại đồ vật và phân tích ảnh vệ tinh. Tuy có nhiều ứng dụng tiềm năng nhưng trong thực tế bài toán Texture Classification lại chưa được áp dụng rộng rãi vì còn nhiều hạn chế. Lý do chính là trong thực tế kết cấu (texture) của cùng một vật thể rất đa dạng về kích cỡ, góc xoay và điều kiện ánh sáng.

Bài toán Texture Classification được chia thành hai giai đoạn chính là giai đoạn trích xuất đặc trưng và giai đoạn phân loại. Trong hai giai đoạn thì giai đoạn rút trích đặc trưng được chú ý và tập trung nghiên cứu nhiều hơn, vì đây là giai đoạn cần những thuật toán mạnh mẽ để rút trích ra những đặc trưng thật hữu ích cho việc phân loại các ảnh thuộc các lớp khác nhau. Đã có rất nhiều thuật toán rút trích đặc trưng được phát minh để có thể phân biệt được những texture có kích cỡ, góc xoay và điều kiện ánh sáng khác nhau. Trong đó, hướng tiếp cận phổ biến nhất là dùng các thuật toán rút trích đặc trưng cục bộ và những kỹ thuật phân tích thống kê.

Trong những năm qua, Texture Classification được nghiên cứu rất rộng rãi, đã có rất nhiều hướng tiếp cận khác nhau được đề xuất. Trong đó, Local Binary Pattern (LBP) [2] được biết đến là một trong những phương pháp thống kê thành công nhất, nhờ vào sự hiệu quả trong việc phân biệt những texture có độ sáng khác nhau mà chỉ sử dụng rất ít tài nguyên tính toán. LBP được dịch là "Mẫu nhị phân địa phương", được Timo Ojala và đồng nghiệp phát minh vào năm 2002. Sau đó rất nhiều biến thể LBP đã được đề xuất và áp dụng để giải quyết các bài toán như Texture Classification, nhận diện gương mặt, ước tính tuổi qua gương mặt.

Trong những năm gần đây, lĩnh vực học sâu (Deep Learning) đã và đang phát triển rất mạnh mẽ. Các mô hình Deep Learning được sử dụng trong nhiều bài toán khác nhau và đạt được kết quả tốt hơn so với những thuật toán trước đó. Theo các LeCun, Bengio và Hinton, "Học sâu cho phép các mô hình tính toán gồm nhiều tầng xử lý để học biểu diễn dữ liệu với nhiều mức trừu tượng khác nhau". Mạng neuron tích chập (Convolutional Neural Network còn được viết tắt là CNN) là một trong những loại mô hình Deep Learning chính. Các mô hình CNN được sử dụng chủ yếu để xử lý các bài toán về thị giác máy tính. Một số ứng dụng tiêu biểu của các mô hình CNN có thể kể đến như là phát hiện các loại bệnh trong y học, phát triển hệ thống xe tự lái, xác định tư thế của người trong ảnh và các bài toán phân loại đối

tượng.

Sau thời gian nghiên cứu, nhóm đã đề xuất được hai loại đặc trưng mới và một hướng giải quyết mới cho bài toán phân loại texture. Cấu trúc của khoá luận được tổ chức như sau:

- Chương 1 mở đầu bằng việc giới thiệu một số khái niệm cơ bản liên quan đến bài toán Texture Classification. Tiếp theo, LBP được trình bày và phân tích những điểm mạnh, điểm yếu, sau đó một số biến thể LBP như LBP Uniform, Rotated LBP, Complete LBP (CLBP), Local Ternary Pattern (LTP) được giới thiệu. Cuối cùng, một số thuật toán học máy được trình bày.
- Chương 2 bắt đầu bằng việc tóm tắt về quá trình hình thành và phát triển của học sâu. Sau đó, một số khái niệm và thành phần cơ bản của mạng neuron tích chập được trình bày.
- Chương 3 giới thiệu phương pháp đề xuất của khoá luận gồm quá trình tiền xử lý dữ liệu, giới thiệu về các đặc trưng và mô hình được đề xuất.
- Chương 4 so sánh và đánh giá kết quả của phương pháp đề xuất với các phương pháp trước đó trên bốn tập dữ liệu texture là New-BarkTex, Outex-TC00013, USPTex và STEX.
- Chương 5 phân tích một số đóng góp của khoá luận và đưa ra một số hướng phát triển tiếp theo của khoá luận.

Chương 1

Cơ sở lý thuyết

1.1 Một số khái niệm cơ bản

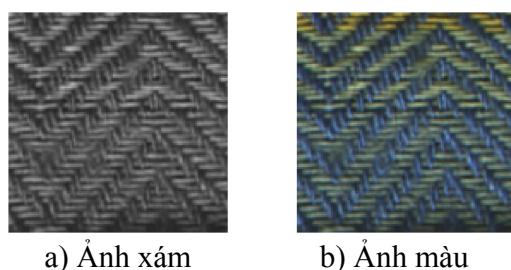
1.1.1 Ảnh đa mức xám (Grayscale Image)

Là một ảnh được biểu diễn bằng một ma trận điểm ảnh (Pixel), mỗi điểm ảnh có giá trị từ 0 đến 255 và được biểu diễn bằng một cấp độ xám tương ứng từ đen đến trắng. Ảnh xám được minh họa trong hình (1.1).

1.1.2 Ảnh màu

Ảnh màu theo lý thuyết của Thomas Young và Hermann Helmholtz là ảnh mà mỗi điểm ảnh là một tổ hợp từ 3 màu cơ bản là màu đỏ (red - R), màu xanh lá cây (green - G) và màu xanh lam (blue - B) viết tắt là ảnh RGB. Hệ màu RGB được phát triển dựa vào cơ chế sinh học của con người, do mắt người có ba loại tế bào cảm quang nhạy cảm với ánh sáng màu đỏ, màu xanh lá cây và màu xanh lam.

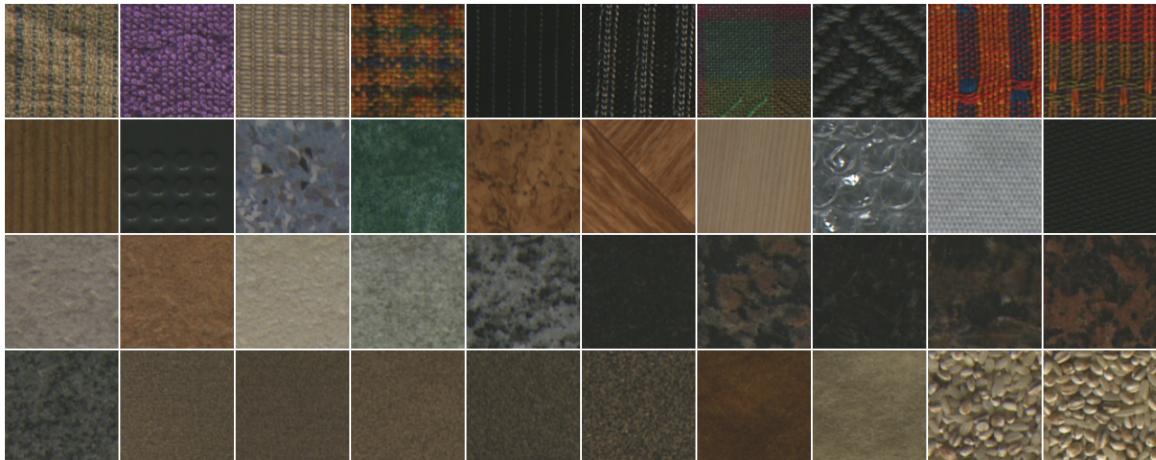
Ảnh RGB được biểu diễn tương tự như ảnh đa mức xám, nhưng mỗi điểm ảnh của ma trận có ba giá trị nằm trong phạm vi từ 0 đến 255. Ba giá trị này được dùng để biểu diễn màu đỏ, màu xanh lá cây và màu xanh lam.



Hình 1.1: Hình minh họa ảnh xám và ảnh màu

1.1.3 Định nghĩa kết cấu (Texture)

Texture là một khái niệm được dùng để chỉ các tính chất có liên quan đến đặc điểm bề mặt của vật thể như cấu tạo, mật độ, sự sắp xếp của bề mặt của vật thể. Dưới đây là một số hình ảnh minh họa texture của một số loại vải, gạch và một số vật liệu khác trong bộ dữ liệu Outex-TC00013.



Hình 1.2: Một số hình trong bộ dữ liệu Outex-TC00013

1.1.4 Định nghĩa phân loại kết cấu (Texture Classification)

Texture classification là một trong số những lĩnh vực quan trọng trong xử lý ảnh. Dữ liệu đầu vào của bài toán là một ảnh texture của một vật thể đã được định nghĩa. Yêu cầu của bài toán là xác định đúng lớp của ảnh texture đó.

1.2 Local Binary Pattern (LBP)

1.2.1 Định nghĩa

LBP được dịch là "Mẫu nhị phân địa phương", được Timo Ojala và đồng nghiệp phát minh vào năm 2002 [2]. LBP là một thuật toán rút trích đặc trưng cục bộ được sử dụng để giải quyết rất nhiều bài toán liên quan đến phân loại texture. Thuật toán LBP sử dụng giá trị của điểm ảnh ở trung tâm làm ngưỡng từ đó phân ngưỡng giá trị của P điểm ảnh xung quanh bán kính R .

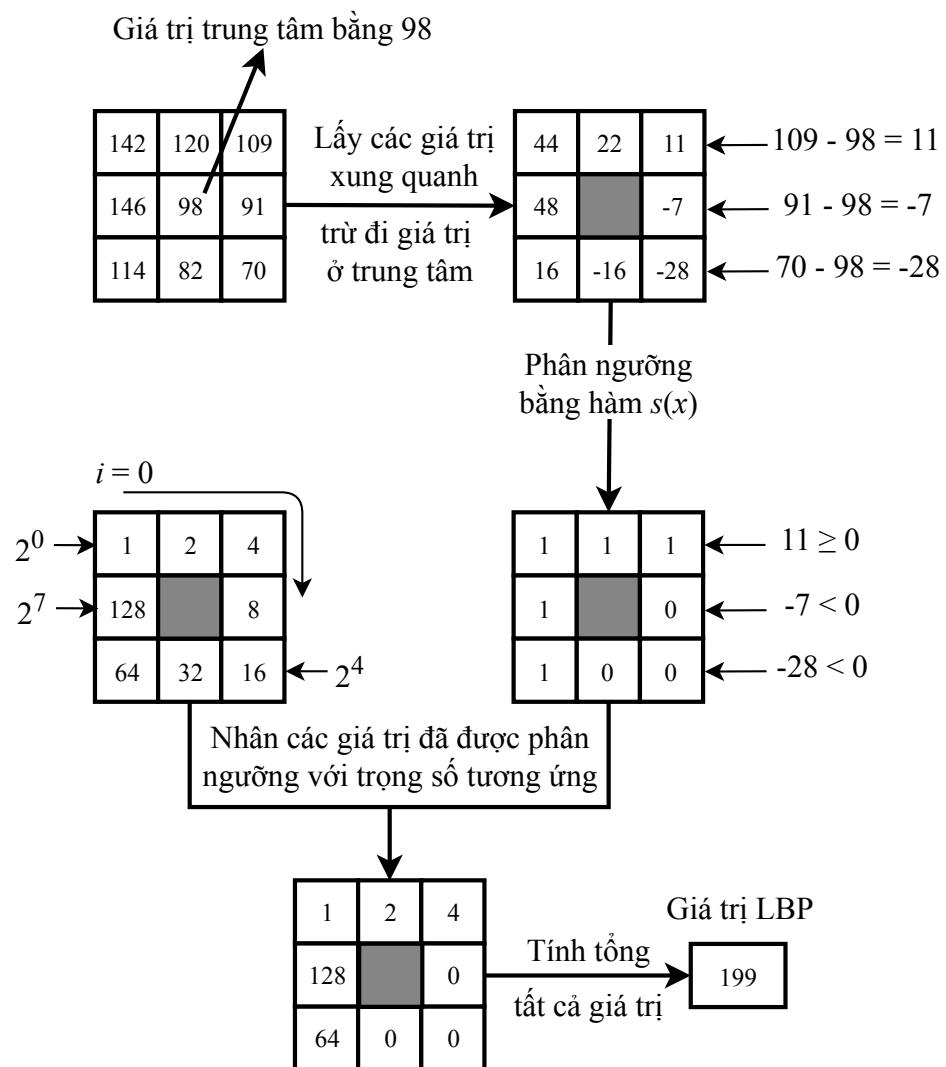
Giá trị của LBP được tính bằng công thức sau:

$$\text{LBP}_{P,R} = \sum_{i=0}^{P-1} s(g_i - g_c) \times 2^i \quad (1.1)$$

Trong đó g_c là giá trị của điểm ảnh ở trung tâm, $\{g_i\}_{i=0}^{P-1}$ là giá trị của P điểm ảnh xung quanh và hàm phân ngưỡng $s(x)$ được định nghĩa như sau:

$$s(x) = \begin{cases} 1 & \text{nếu } x \geq 0 \\ 0 & \text{nếu } x < 0 \end{cases} \quad (1.2)$$

Dưới đây là hình minh họa quá trình trích xuất giá trị LBP_{8,1}.



Hình 1.3: Hình minh họa quá trình trích xuất giá trị LBP_{8,1}

1.2.2 Ưu điểm và nhược điểm

Ưu điểm:

- Thuật toán đơn giản, hiệu quả.
- Độ phức tạp tính toán của thuật toán thấp.
- Ít bị ảnh hưởng bởi việc thay đổi độ sáng.

Nhược điểm:

- Không hiệu quả đối với ảnh xoay.
- Độ phức tạp tính toán tăng theo cấp số mũ nếu tăng số điểm mẫu.
- Bị mất đi nhiều thông tin quan trọng khi phân ngưỡng các điểm mẫu.

1.2.3 Một số biến thể LBP

Từ khi ra đời đến nay, LBP đã được nghiên cứu rất rộng rãi và đã có rất nhiều biến thể LBP được đề xuất. Dưới đây khoá luận trình bày một số biến thể LBP thường được sử dụng.

1.2.3.1 LBP Uniform

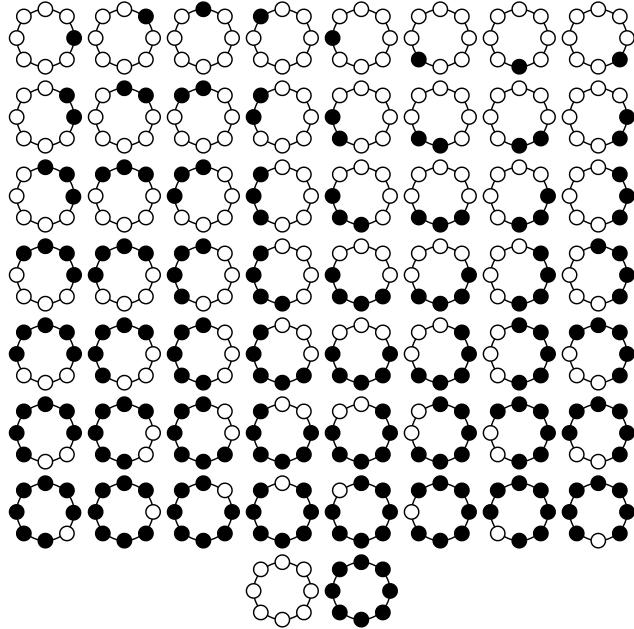
LBP Uniform được được Timo Ojala và đồng nghiệp đề xuất [3], cách tính giá trị của LBP Uniform tương tự như LBP thông thường, tuy nhiên chỉ những giá trị LBP thoả điều kiện đặt ra mới được giữ lại để thống kê, tất cả các giá trị LBP không thoả sẽ được gán bằng một giá trị định trước. Điều kiện của LBP Uniform là không có nhiều hơn 2 transition trong dãy các giá trị đã phân ngưỡng. Một transition là một cặp gồm một bit 0 và một bit 1 đứng cạnh nhau, ví dụ dãy 01000000 có 2 transition và là một LBP Uniform, dãy 11001001 có 4 transition và không phải là một LBP Uniform.

Theo thống kê của bài báo [3], các giá trị LBP Uniform chiếm trên 90% số lượng các giá trị LBP trong ảnh mặt người và các ảnh liên quan đến Texture, nhờ đó LBP Uniform giúp giảm độ phức tạp tính toán nhưng vẫn giữ được phần lớn thông tin quan trọng.

Có tổng cộng 58 giá trị LBP Uniform đối với $LBP_{8,1}$ là: 0, 1, 2, 3, 4, 6, 7, 8, 12, 14, 15, 16, 24, 28, 30, 31, 32, 48, 56, 60, 62, 63, 64, 96, 112, 120, 124, 126, 127, 128, 129, 131, 135, 143, 159, 191, 192, 193, 195, 199, 207, 223, 224, 225, 227, 231, 239, 240, 241, 243, 247, 248, 249, 251, 252, 253, 254, 255. Các giá trị không nằm trong 58 giá trị trên sẽ được gán vào cùng một loại để thống kê.

1.2.3.2 Local Ternary Pattern (LTP)

LTP là một biến thể của LBP được đề xuất bởi Xiaoyang Tan và Bill Triggs [4] nhằm giữ lại thêm thông tin từ sự khác biệt giữa các điểm ảnh xung quanh và điểm



Hình 1.4: Hình các LBP Uniform

ảnh trung tâm. LBP chỉ phân ngưỡng sự khác biệt thành bit 1 hoặc bit 0, thay vào đó hàm phân ngưỡng $s_t(x, c)$ của LTP phân thành ba ngưỡng khác nhau là 1, 0 và -1 bằng công thức sau:

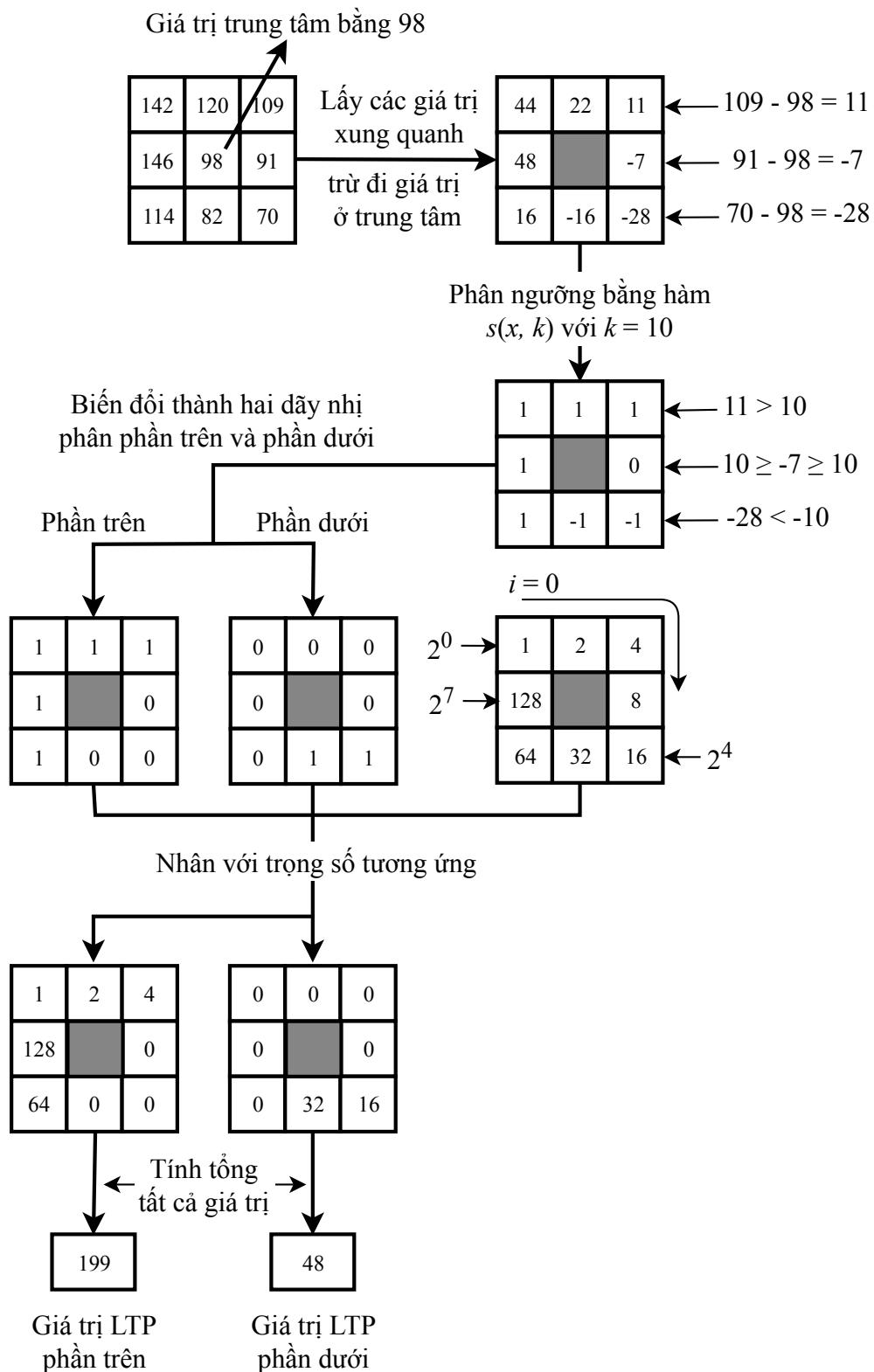
$$s_t(g_i - g_c, c) = \begin{cases} 1 & \text{nếu } g_i - g_c > c \\ -1 & \text{nếu } g_i - g_c < c \\ 0 & \text{nếu } c \geq g_i - g_c \geq -c \end{cases} \quad (1.3)$$

Trong đó c là giá trị phân ngưỡng được định nghĩa tùy thuộc vào từng trường hợp.

Sau đó dãy kết quả sau khi phân ngưỡng sẽ được biến đổi thành hai dãy mã nhị phân khác nhau để tính hai giá trị phần trên và phần dưới của LTP. Các số -1 trong mảng kết quả S_t sẽ được biến đổi thành 0 và lưu vào mảng A_u , để có được mảng A_l các số 1 trong mảng S_t sẽ được đổi thành số 0 và các số -1 sẽ được đổi thành số 1. Sau cùng, giá trị phần trên và phần dưới của LTP sẽ được tính độc lập trên mảng A_u và A_l bằng công thức sau:

$$\text{LTP}_{P,R} = \sum_{i=0}^{P-1} A_i \times 2^i \quad (1.4)$$

Dưới đây là hình minh họa quá trình trích xuất hai giá trị LTP_{8,1}.



Hình 1.5: Hình minh họa quá trình trích xuất hai giá trị LTP_{8,1}

1.2.3.3 Rotated LBP (RLBP)

Thuật toán LBP thông thường không giữ được thông tin về độ chênh lệch giá trị giữa các điểm ảnh, do hàm phân ngưỡng. Một vấn đề khác là LBP thông thường không hoạt động tốt trên ảnh xoay. Do đó, Rakesh Mehta và Karen Egiazarian đã đề xuất RLBP [5] để xử lý vấn đề ảnh bị xoay và giữ lại một ít thông tin về độ chênh lệch giữa các điểm ảnh.

Để tính giá trị của RLBP, bước đầu tiên là xác định vị trí D của điểm mẫu chính, đây là điểm mẫu có độ chênh lệch lớn nhất đối với giá trị của điểm ảnh ở giữa. Công thức tìm vị trí D được định nghĩa như sau:

$$D = \text{agrmax} | g_i - g_c | \quad (1.5)$$

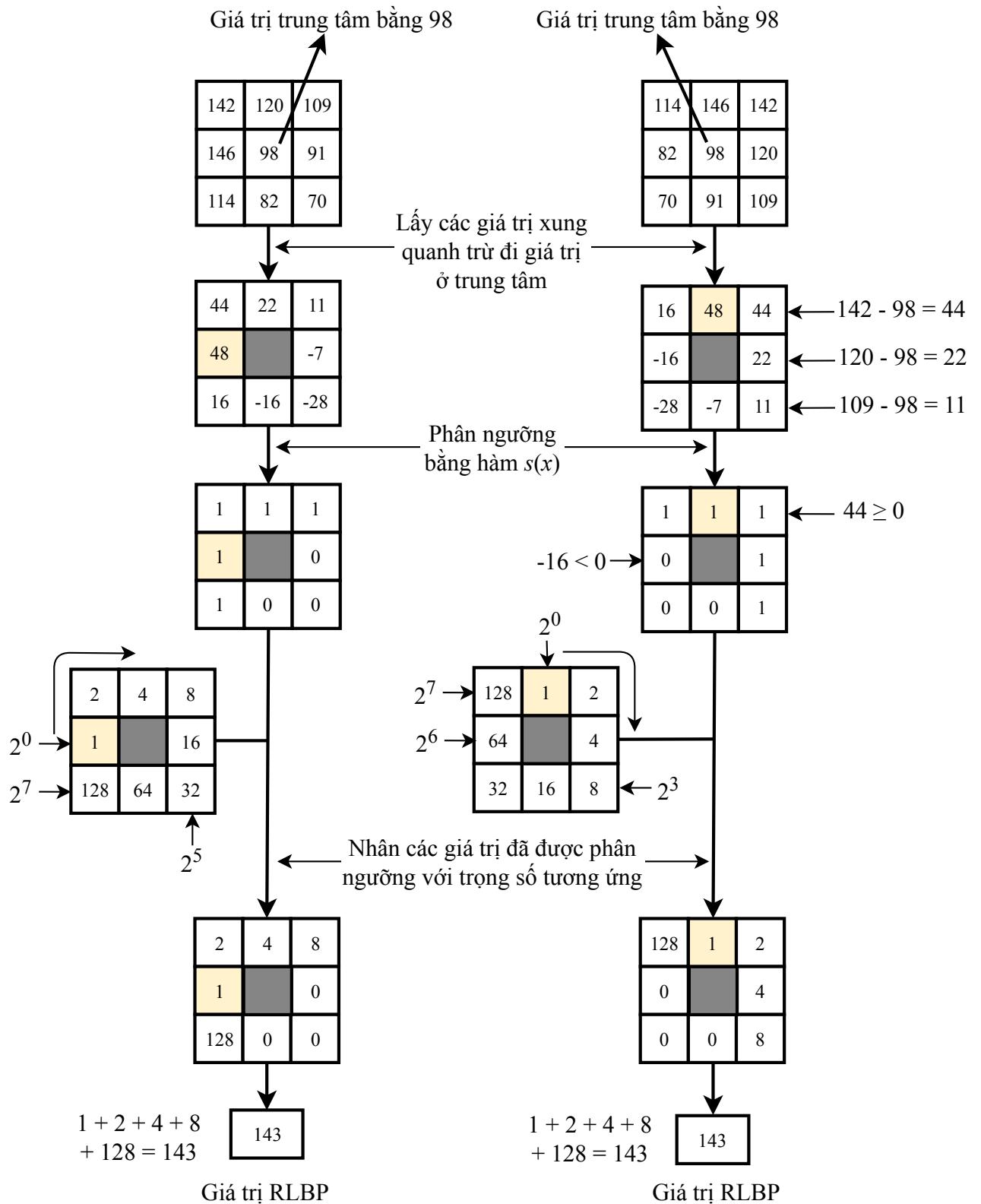
Trong đó g_c là giá trị của điểm ảnh ở trung tâm, $\{g_i\}_{i=0}^{P-1}$ là giá trị của P điểm ảnh xung quanh và hàm $\text{agrmax}(x)$ là hàm trả về vị trí của giá trị lớn nhất trong mảng x .

Công thức dùng để tính giá trị RLBP được định nghĩa như sau:

$$\text{RLBP}_{P,R} = \sum_{i=0}^{P-1} s(g_i - g_c) \times 2^{\text{mod}(i-D,P)} \quad (1.6)$$

Trong đó hàm $\text{mod}(x)$ là hàm chia lấy phần dư và $s(x)$ là hàm phân ngưỡng (1.2).

Dưới đây là hình minh họa quá trình trích xuất giá trị RLBP_{8,1}. Trong hình, hai giá trị RLBP được trích xuất để cho thấy sự hiệu quả của RLBP đối với ảnh bị xoay. Dù đã bị xoay 90° theo hướng kim đồng hồ nhưng hai giá trị RLBP được trích xuất ở hai điểm vẫn có cùng giá trị. Ô được tô màu vàng nhạt đánh dấu giá trị có độ chênh lệch lớn nhất so với trung tâm, các trọng số được gán từ vị trí này trở đi theo hướng kim đồng hồ.



Hình 1.6: Hình minh họa quá trình trích xuất hai giá trị $RLBP_{8,1}$

1.2.3.4 Complete LBP (CLBP)

CLBP được phát minh bởi Zhenhua Guo và đồng nghiệp vào năm 2010 [6]. CLBP được đề xuất nhằm giữ lại một phần những thông tin có ích như giá trị của điểm ảnh trung tâm, độ khác nhau giữa các điểm ảnh và thông tin toàn cục của bức ảnh. CLBP gồm ba thành phần là CLBP_S, CLBP_M và CLBP_C. Trong đó, CLBP_S được tính tương tự như LBP thông thường bằng công thức (1.1). CLBP_M được tính dựa trên độ khác nhau giữa P điểm ảnh so với giá trị trung tâm g_c . Độ khác nhau giữa các điểm ảnh $\{g_i\}_{i=0}^{P-1}$ so với trung tâm được tính bằng công thức:

$$d_i = |g_i - g_c| \quad (1.7)$$

CLBP_M được tính bằng công thức:

$$\text{CLBP_M}_{P,R} = \sum_{i=0}^{P-1} s(d_i, c) \times 2^i \quad (1.8)$$

Trong đó c là giá trị phân ngưỡng được định nghĩa tùy thuộc vào từng trường hợp, hàm phân ngưỡng $s(x, c)$ được định nghĩa:

$$s(x, c) = \begin{cases} 1 & \text{nếu } x \geq c \\ 0 & \text{nếu } x < c \end{cases} \quad (1.9)$$

CLBP_C được tính dựa vào giá trị của điểm ảnh ở giữa g_c và giá trị trung bình của toàn bộ điểm ảnh trong bức ảnh c_m . Công thức tính CLBP_C được định nghĩa như sau:

$$\text{CLBP_M} = s(g_c, c_m) \quad (1.10)$$

Trong đó hàm phân ngưỡng $s(x, c)$ được tính bằng công thức (1.9)

1.3 Học máy (Machine Learning)

1.3.1 Định nghĩa

Trong lĩnh vực trí tuệ nhân tạo có một nhánh nghiên cứu về phát triển khả năng tự học của máy tính được gọi là học máy (Machine Learning). Machine Learning là một ngành nghiên cứu về những thuật toán, những mô hình thống kê để máy tính có thể thực hiện một công việc cụ thể mà không cần phải lập trình một cách chi tiết từng bước. Các thuật toán Machine Learning được phân ra làm 2 loại chính là:

- Học có giám sát (Supervised Learning) là phương pháp sử dụng những dữ liệu đầu vào đã được con người gán nhãn để tạo ra một hàm, sao cho hàm này có khả năng xử lý những dữ liệu đầu vào mới và xác định được giá trị đầu ra chính xác. Ví dụ cho dữ liệu đầu vào là một số hình ảnh con chó và một số hình con mèo đã được phân thành hai loại, thuật toán phải xác định những hình ảnh chưa từng gặp là mèo hay là chó.
- Học không giám sát (Unsupervised Learning) là phương pháp sử dụng những dữ liệu đầu vào chưa được con người gán nhãn hay phân loại. Nhiệm vụ của thuật toán là phải học từ những dữ liệu này và phân loại chúng thành một số nhóm đã được định trước. Ví dụ cho dữ liệu đầu vào là một số hình ảnh con chó và một số hình con mèo chưa được phân loại, thuật toán phải xác định những hình ảnh nào là mèo, những hình ảnh nào là chó.

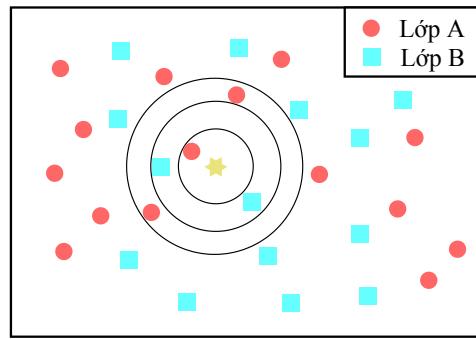
1.3.2 Một số thuật toán

Trong phần này khoá luận sẽ trình bày về một số thuật toán Machine Learning thường được sử dụng.

1.3.2.1 K-Nearest Neighbors (KNN)

K-Nearest Neighbors là một trong số những thuật toán Supervised Learning đơn giản nhất và thường được sử dụng trong những bài toán về Texture Classification. Để phân loại một ảnh mới bằng thuật toán KNN, bước đầu tiên là tìm K ảnh đã được gán nhãn gần nhất trong không gian vector đặc trưng, sau đó ảnh mới được gán là lớp chiếm tỉ lệ cao nhất trong K ảnh vừa tìm được. Trong đó K là hằng số được người dùng định nghĩa. Trong trường hợp K bằng 1, ảnh cần được phân loại sẽ được gán là lớp của ảnh huấn luyện gần nhất trong không gian vector đặc trưng. Giá trị K tối ưu sẽ thay đổi theo từng bài toán.

Dưới đây hình minh họa về việc phân lớp một điểm dữ liệu bằng thuật toán KNN. Trong ví dụ có hai lớp là lớp A (hình tròn đỏ) và lớp B (hình vuông xanh), điểm dữ liệu cần phân lớp là hình ngôi sao 6 cánh màu vàng.



Hình 1.7: Hình minh họa thuật toán KNN

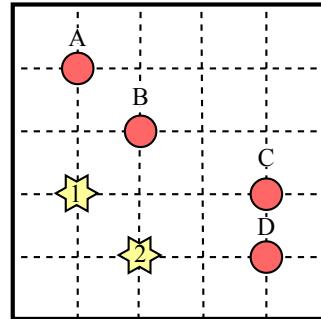
Trong hình (1.7) ta có thể thấy điểm dữ liệu có thể thuộc về các lớp khác nhau nếu tham số K thay đổi. Trong trường hợp trên, nếu K bằng 1 hoặc 5 thì điểm dữ liệu thuộc về lớp A và khi K bằng 3 thì điểm dữ liệu thuộc về lớp B.

1.3.2.2 K -Means Clustering

K -Means Clustering là một trong số những thuật toán Unsupervised Learning đơn giản nhất. Để gom nhóm dữ liệu, thuật toán K -Means Clustering sẽ khởi tạo ngẫu nhiên K điểm trung tâm của từng lớp trong không gian vector đặc trưng của tập dữ liệu. Sau đó thuật toán được thực hiện với ba bước:

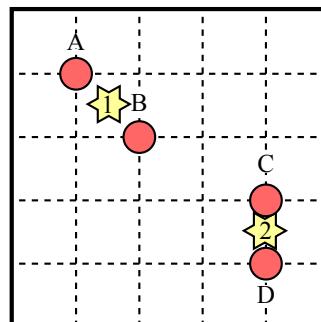
- Bước 1: Các điểm dữ liệu có khoảng cách gần với điểm trung tâm lớp nào sẽ được gán bằng lớp đó.
- Bước 2: Toạ độ của các điểm trung tâm lớp sẽ được gán bằng toạ độ trung bình của những điểm thuộc lớp đó.
- Bước 3: Lặp lại từ bước 1 cho đến khi không có sự thay đổi lớp của các điểm dữ liệu ở bước 1.

Dưới đây là ví dụ phân lớp tập dữ liệu gồm bốn điểm dữ liệu A, B, C, D thành hai lớp bằng thuật toán K-Means Clustering. Đầu tiên hai điểm trung tâm của hai lớp được khởi tạo ngẫu nhiên.



Hình 1.8: Minh họa trạng thái khởi tạo của thuật toán K-Means Clustering

Trong hình (1.8), các điểm dữ liệu được ký hiệu bằng hình tròn đỏ và các điểm trung tâm lớp được ký hiệu bằng hình sao 6 cánh màu vàng. Sau khi thực hiện bước 1, điểm A, B thuộc về lớp 1 và điểm C, D thuộc về lớp 2. Dưới đây là hình trung tâm của hai lớp đã được dịch chuyển đến tọa độ trung bình của các điểm thuộc lớp.

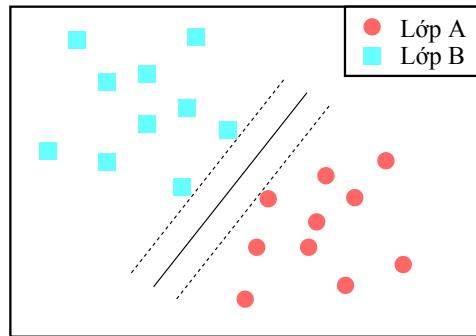


Hình 1.9: Cập nhật các điểm trung tâm trong thuật toán K-Means Clustering

Thuật toán được ngừng lại do không có sự thay đổi lớp của các điểm dữ liệu khi thực hiện bước 1. Như vậy tập dữ liệu gồm bốn điểm dữ liệu đã được phân lớp bằng thuật toán K-Means Clustering. Kết quả là điểm dữ liệu A và B thuộc lớp 1, điểm dữ liệu C và D thuộc lớp 2.

1.3.2.3 Máy vector hỗ trợ (Support Vector Machine - SVM)

Thuật toán SVM là một thuật toán học có giám sát được phát minh bởi Corinna Cortes và Vladimir Vapnik [7]. SVM là một thuật toán phân loại nhị phân (chỉ phân loại dữ liệu thành hai lớp khác nhau). Thuật toán SVM phân loại hai lớp bằng cách tìm ra một siêu phẳng (một mặt phẳng n chiều) phân cách hai lớp, sao cho khoảng cách giữa siêu phẳng đó đến điểm dữ liệu gần nhất của mỗi lớp là cực đại.



Hình 1.10: Minh họa phân loại hai lớp bằng thuật toán SVM

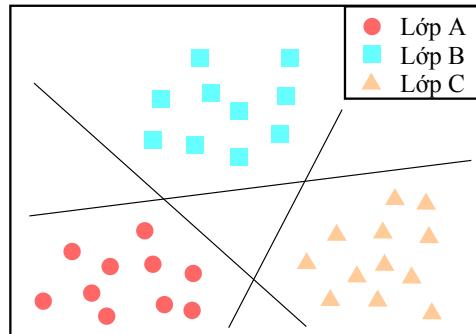
Thuật toán SVM ban đầu chỉ có thể phân loại dữ liệu một cách tuyến tính, nhưng sau đó được mở rộng để có thể phân loại phi tuyến tính bằng cách ánh xạ dữ liệu vào một không gian nhiều chiều hơn. Trong không gian nhiều chiều hơn SVM chỉ phân loại một cách tuyến tính, nhưng ranh giới của nó trên không gian của dữ liệu thì nó phân loại một cách phi tuyến tính.

SVM ban đầu chỉ là một thuật toán phân loại nhị phân, nhưng trong thực tế thường gặp những bài toán yêu cầu phân loại thành nhiều lớp hơn. Do đó, hai chiến lược đã được phát triển để có thể dùng SVM cho bài toán phân loại đa lớp là chiến thuật Một so với tất cả (one-versus-all) và Một so với một (one-versus-one).

Chiến thuật Một so với một: Mỗi cặp lớp khác nhau sẽ có một SVM khác nhau phân cách. Lớp của một điểm dữ liệu mới được gán bằng lớp được các SVM bầu chọn nhiều nhất.

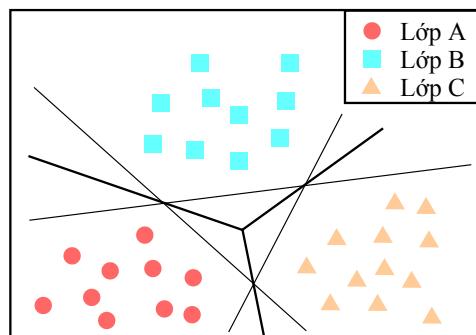
Chiến thuật Một so với tất cả: Mỗi lớp sẽ có một SVM phân cách lớp đó với tất cả các lớp khác. Một điểm dữ liệu mới được gán lớp khi điểm dữ liệu đó chỉ thuộc về một lớp và không thuộc bất cứ lớp nào khác.

Dưới đây là ví dụ minh họa phân loại ba lớp dùng SVM với chiến thuật Một so với tất cả.



Hình 1.11: Minh họa SVM phân loại nhiều lớp với chiến thuật Một so với tất cả cơ bản

Ta có thể thấy trong hình (1.11) vẫn có nhiều vùng có thể thuộc nhiều lớp hoặc không thuộc lớp nào. Do đó, hàm trả về kết quả của các SVM sẽ được điều chỉnh để cho ra một giá trị có thể so sánh, lớp của điểm dữ liệu sẽ thuộc về lớp nào có giá trị trả về lớn nhất. Dưới đây là hình minh họa về việc áp dụng thuật toán này.



Hình 1.12: Minh họa SVM phân loại nhiều lớp với chiến thuật Một so với tất cả

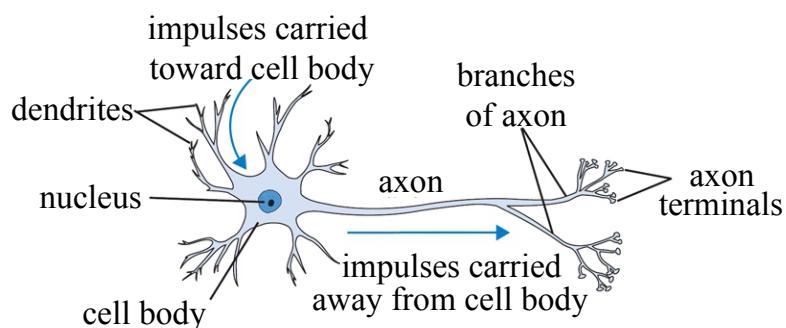
Dựa vào việc áp dụng thuật toán trên các điểm dữ liệu nằm trong những vùng thuộc nhiều lớp hoặc không thuộc lớp nào có thể được phân loại.

Chương 2

Mạng neuron nhân tạo (Artificial Neural Networks)

2.1 Định nghĩa

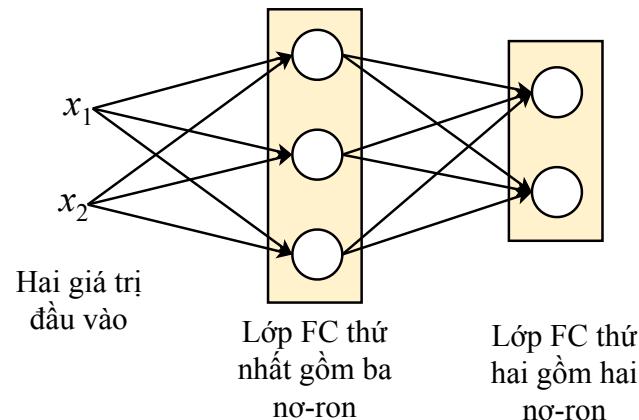
Mạng neuron nhân tạo (Artificial Neural Networks - ANN) là một nhánh con trong ngành Machine Learning, được lấy ý tưởng từ bộ não của con người. Hình (2.1) minh họa một neuron sinh học, mỗi neuron sinh học nhận tín hiệu từ nhiều sợi nhánh (dendrite) và trả về một tín hiệu đầu ra ở sợi trực (axon), mỗi sợi trực chia ra nhiều nhánh con kết nối với các sợi nhánh của các neuron khác để truyền tín hiệu. Lấy ý tưởng từ neuron sinh học, neuron nhân tạo được mô hình hoá bằng một hàm tính toán gồm nhiều dữ liệu đầu vào, mỗi dữ liệu đầu vào có tầm quan trọng khác nhau được quyết định bởi các tham số w . Các tham số w có thể được điều chỉnh qua quá trình huấn luyện. Để tạo ra tín hiệu đầu ra, neuron sinh học sẽ lấy tổng tín hiệu từ các sợi nhánh, nếu tín hiệu vượt qua một ngưỡng nào đó, neuron sẽ trả về một tín hiệu đầu ra ở sợi trực. Trong mô hình tính toán của neuron nhân tạo, quá trình đó được thể hiện bằng một hàm kích hoạt (Activation Function).



Hình 2.1: Hình minh họa một neuron sinh học (nguồn <http://cs231n.github.io/>)

Để tạo ra một mạng neuron nhân tạo, các neuron nhân tạo thường được sắp xếp thành từng lớp, một loại lớp thường được dùng là lớp Fully-connected (FC). Trong

mỗi lớp FC có một hoặc nhiều neuron nhân tạo, các neuron trong một lớp FC liên kết từng đôi một với các neuron trong lớp FC liền kề và các neuron trong cùng một lớp không kết nối với nhau.



Hình 2.2: Hình minh họa một mạng neuron nhân tạo gồm 2 lớp FC

2.2 Một số thành phần quan trọng

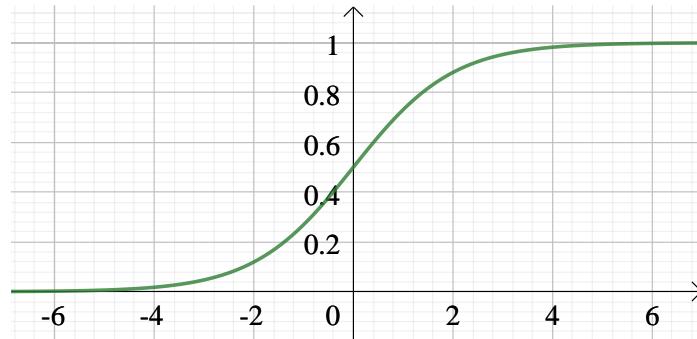
2.2.1 Hàm Sigmoid

Hàm Sigmoid là một hàm kích hoạt không tuyến tính. Trong quá khứ, hàm Sigmoid được sử dụng rất rộng rãi do có cách hoạt động khá giống cơ chế xử lý tín hiệu của neuron sinh học. Đầu vào của hàm Sigmoid là một số thực, hàm sẽ trả về một số nằm trong khoảng từ 0 đến 1 bằng công thức:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.1)$$

Trong đó e là cơ số của logarit tự nhiên có giá trị xấp xỉ 2,71828.

Gần đây, hàm Sigmoid ít được sử dụng vì độ dốc (gradient) của hàm tại gần 0 và 1 quá nhỏ, điều này làm cho quá trình huấn luyện bằng thuật toán lan truyền ngược hoạt động không tốt. Dưới đây là hình đồ thị của hàm Sigmoid.



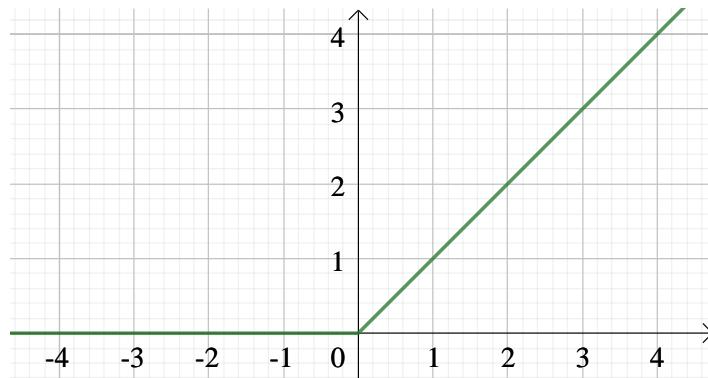
Hình 2.3: Đồ thị hàm Sigmoid

2.2.2 Hàm Rectified Linear Units (ReLU)

Trong những năm gần đây, hàm kích hoạt ReLU được sử dụng rất phổ biến. Hàm ReLU thực chất là một hàm phân nguồng tại 0, các giá trị nhỏ hơn 0 sẽ được gán bằng 0, những giá trị lớn hơn 0 sẽ được giữ nguyên giá trị. Công thức của hàm ReLU:

$$f(x) = \max(0, x) \quad (2.2)$$

Những điểm mạnh của hàm ReLU là độ phức tạp tính toán rất thấp và hoạt động tốt với thuật toán lan truyền ngược. Dưới đây là hình đồ thị của hàm ReLU.



Hình 2.4: Đồ thị hàm ReLU

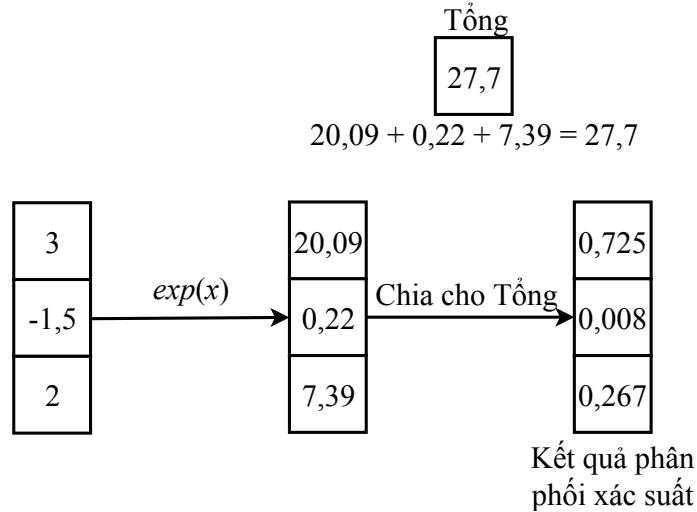
2.2.3 Hàm Softmax

Hàm Softmax là hàm được sử dụng để phân phối xác suất dựa vào dãy kết quả đầu ra của mạng neuron nhân tạo. Công thức của hàm Softmax dùng để tính P_i (xác suất của lớp i) từ giá trị x_i thuộc dãy kết quả gồm n giá trị là:

$$P_i = \frac{\exp(x_i)}{\sum_{j=0}^{n-1} \exp(x_j)} \quad (2.3)$$

Trong đó hàm $\exp(x)$ là hàm lũy thừa cơ số e , công thức là $\exp(x) = e^x$.

Dưới đây là ví dụ minh họa sử dụng hàm Softmax để phân phối xác suất cho ba giá trị.



Hình 2.5: Phân phối xác suất bằng hàm Softmax

2.2.4 Hàm mất mát (Loss Function)

Hàm mất mát còn được gọi là Loss Function hoặc Cost Function được dùng để đánh giá kết quả của một dự đoán. Hàm mất mát sẽ trả về một số thực không âm tỉ lệ thuận với độ trầm trọng của sai sót. Hàm mất mát thường được sử dụng nhất trong các mạng neuron nhân tạo là hàm Cross-Entropy Loss.

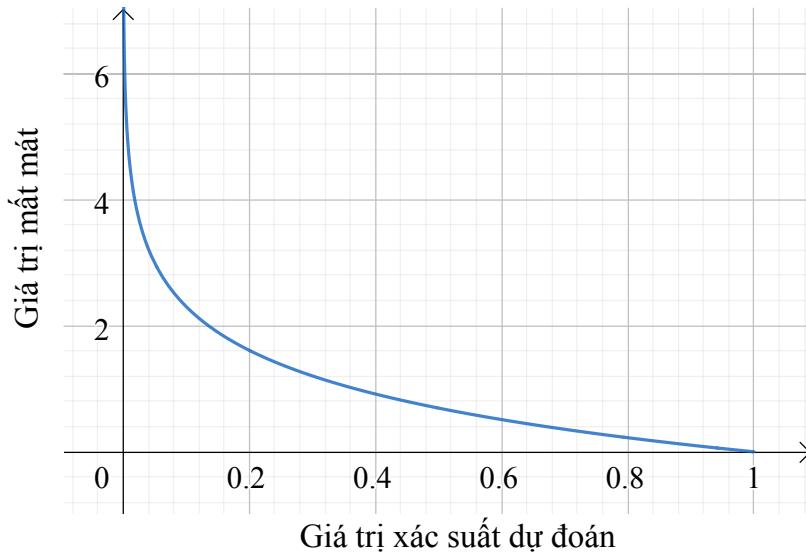
2.2.4.1 Hàm Cross-Entropy Loss

Hàm Cross-Entropy Loss còn được gọi là hàm Log Loss, được dùng để đánh giá kết quả của một mạng neuron nhân tạo có kết quả đầu ra dạng phân phối xác suất. Công thức được sử dụng để đánh giá một kết quả phân phối xác suất p gồm n giá trị được định nghĩa như sau:

$$L = - \sum_{i=0}^{n-1} y_i \log(p_i) \quad (2.4)$$

Trong đó $\{y_i\}_{i=0}^{n-1}$ là kết quả phân phối xác suất yêu cầu và hàm $\log(x)$ là hàm logarit tự nhiên (có cơ số là e).

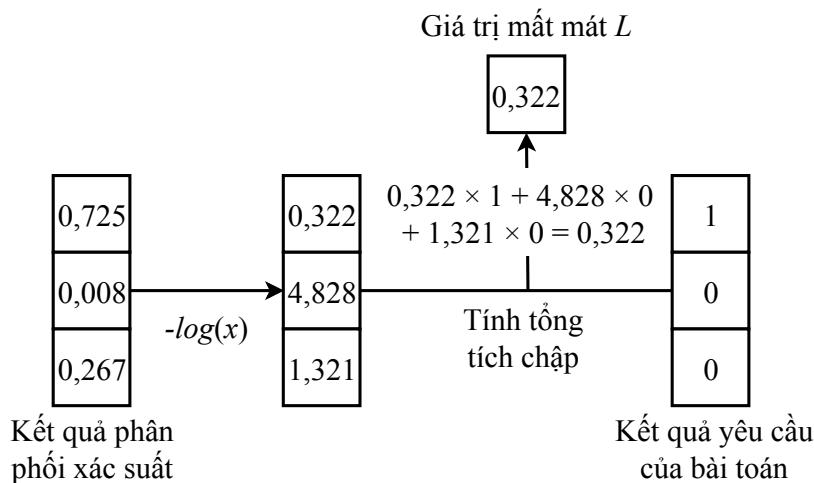
Dưới đây là đồ thị của hàm Log Loss.



Hình 2.6: Đồ thị hàm Log Loss

Có thể thấy từ hình (2.6), giá trị xác suất dự đoán gần với 1 thì giá trị mất mát L rất nhỏ, nhưng khi giá trị xác suất dự đoán gần với 0 thì giá trị mất mát L rất lớn.

Dưới đây là ví dụ minh họa cách tính giá trị mất mát L từ một kết quả phân phối xác suất có ba giá trị. Kết quả yêu cầu của bài toán là trả về lớp thứ nhất, do đó dãy kết quả yêu cầu có giá trị là 1 ở ô thứ nhất và 0 ở các ô còn lại.



Hình 2.7: Minh họa cách tính giá trị mất mát từ kết quả phân phối xác suất

2.2.5 Thuật toán lan truyền ngược (Backpropagation Algorithm)

Thuật toán lan truyền ngược là thuật toán cốt lõi của quá trình huấn luyện các mô hình neuron nhân tạo. Thuật toán này dùng đạo hàm riêng để tìm ra hướng để

cập nhật các tham số w và b của mô hình dựa vào giá trị mất mát L .

Ví dụ tìm hướng để cập nhật các tham số x, y, z của một hàm $f(x, y, z) = xy + z$ được thực hiện như sau. Đầu tiên hàm được biến đổi thành các phép toán đơn giản $f = m + z$ với $m = xy$. Sau đó các đạo hàm riêng được tính:

$$\frac{\partial f}{\partial m} = 1, \frac{\partial f}{\partial z} = 1, \frac{\partial m}{\partial x} = y, \frac{\partial m}{\partial y} = x$$

Từ đó quy tắc chuỗi (Chain Rule) được sử dụng để tính đạo hàm riêng của f theo x và y như sau:

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial m} \frac{\partial m}{\partial x} = y \text{ và } \frac{\partial f}{\partial y} = \frac{\partial f}{\partial m} \frac{\partial m}{\partial y} = x$$

Đạo hàm riêng của f theo z được ký hiệu $\frac{\partial f}{\partial z}$ là độ ảnh hưởng của z đến f khi z thay đổi, ta có $\frac{\partial f}{\partial z} = 1$ nghĩa là khi tăng z bao nhiêu thì f sẽ tăng lên bấy nhiêu. Như vậy, dựa vào đạo hàm riêng của f theo các tham số x, y, z ta có thể biết hướng để cập nhật các tham số để làm giảm giá trị của hàm f . Tương tự như vậy, thuật toán lan truyền ngược được sử dụng để tìm ra hướng cập nhật các tham số sao cho giá trị mất mát L của mô hình ngày càng giảm đi, từ đó tăng độ chính xác của mô hình.

2.3 Học sâu (Deep Learning)

2.3.1 Định nghĩa

Mô hình học sâu là một mạng neuron nhân tạo có nhiều lớp dùng để biểu diễn dữ liệu dưới nhiều tầng trừu tượng. Các mô hình học sâu có thể được huấn luyện bằng phương pháp giám sát, không giám sát hoặc nửa giám sát. Hai loại mô hình được nghiên cứu rộng rãi nhất trong thời gian gần đây là mạng neuron tích chập (Convolutional Neural Network) và mạng neuron hồi quy (Recurrent Neural Network).

2.3.2 Lịch sử hình thành, phát triển

Giai đoạn Perceptron: Đây là nền móng đầu tiên của Deep Learning. Perceptron là một thuật toán học có giám sát giúp giải quyết bài toán phân lớp nhị phân, được sáng tạo bởi Frank Rosenblatt năm 1957. Trong thời gian đó thuật toán này mang lại nhiều kỳ vọng nhưng nó đã nhanh chóng được chứng minh không thể giải quyết những bài toán đơn giản. Điều này khiến cho việc nghiên cứu về các mạng neuron nhân tạo bị gián đoạn gần 20 năm.

Giai đoạn Perceptron đa lớp: Sau một thời gian dài các công trình nghiên cứu về perceptron bị gián đoạn. Năm 1986, Geoffrey Hinton cùng với hai tác giả khác xuất bản một bài báo khoa học chứng minh rằng một mô hình với nhiều lớp Perceptron có thể được huấn luyện một cách hiệu quả dựa trên thuật toán lan truyền ngược. Mô hình này mang lại một vài thành công ban đầu, nổi trội là mô hình Mạng neuron tích chập để giải quyết bài toán nhận dạng chữ số viết tay. Các mô hình này được

kỳ vọng sẽ giải quyết nhiều bài toán phân loại hình ảnh khác. Tuy nhiên, những mô hình này lại có hai vấn đề không thể khắc phục trong thời điểm đó như cần số lượng ảnh lớn để huấn luyện trong khi máy ảnh kỹ thuật số lại chưa phổ biến, độ phức tạp tính toán cao vượt quá khả năng tính toán của các máy tính thời đó. Những vấn đề này khiến cho các mô hình dần neuron nhân tạo dần được thay thế bởi Máy vector hỗ trợ. Việc nghiên cứu về các mạng neuron nhân tạo một lần nữa bị gián đoạn gần 20 năm.

Giai đoạn ra đời của Deep Learning: Vào năm 2006, Geoffrey Hinton giới thiệu ý tưởng sử dụng tiền huấn luyện không giám sát (Unsupervised Pretraining) thông qua Deep Belief Network (DBN) để huấn luyện các mô hình. Nhờ ý tưởng này, ông đã thành công trong việc huấn luyện những mô hình neuron nhân tạo nhiều lớp hơn. Từ đó, ông gọi các mô hình này là các mô hình học sâu.

Giai đoạn bùng nổ của Deep Learning: Năm 2012, tại cuộc thi thường niên ImageNet Large Scale Visual Recognition Challenge (ILSVRC), Alex Krizhevsky, Ilya Sutskever, và Geoffrey Hinton tham gia và đạt tỷ lệ sai top 5 (top-5 error rate) là 16%, tốt hơn rất nhiều so với kết quả tốt nhất năm trước đó là 26%. Mô hình chiến thắng là một Mạng neuron tích chập, sau này được đặt tên là AlexNet. Sau AlexNet, tất cả các mô hình giành giải cao nhất trong các năm tiếp theo đều là các Mạng neuron tích chập, các mô hình giành giải cao của các năm tiếp theo là: ZFNet (2013), GoogLeNet/Inception (2014), VGGNet (2014), ResNet (2015), Inception-ResNet (2016).

Những yếu tố dẫn đến sự bùng nổ của lĩnh vực học sâu:

- Sự xuất hiện của nhiều bộ dữ liệu có lượng ảnh lớn, chất lượng cao.
- Khả năng tính toán song song với tốc độ cao của các dòng GPU mới.
- Sự ra đời của ReLU và các hàm kích hoạt liên quan làm hạn chế vấn đề của thuật toán lan truyền ngược đổi với những mô hình nhiều lớp.
- Các kiến trúc và các kỹ thuật huấn luyện mới trong lĩnh vực liên tục được nghiên cứu và phát triển.
 - Nhiều thư viện mới hỗ trợ việc xây dựng và huấn luyện các mô hình Deep Learning: tensorflow, pytorch, keras, theano, caffe, mxnet ...
 - Nhiều kỹ thuật tối ưu mới: Adagrad, RMSProp, Adam, SGD ...

2.3.3 Mạng neuron tích chập (Convolutional Neural Network - CNN)

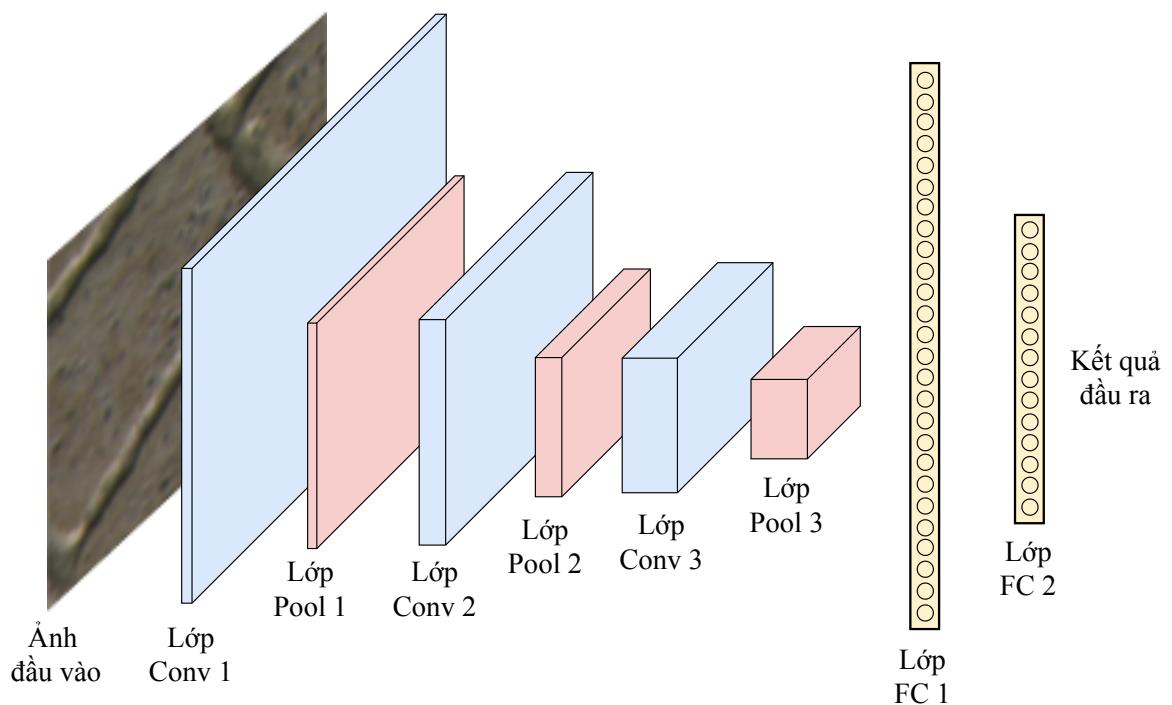
2.3.3.1 Định nghĩa

Mạng neuron tích chập (gọi tắt là CNN) là loại mô hình học sâu giúp giải quyết các bài toán liên quan đến hình ảnh với độ chính xác rất cao, các hệ thống xử lý ảnh

lớn của Google, Facebook hay Amazon đã và đang sử dụng các mô hình CNN vào các sản phẩm của họ.

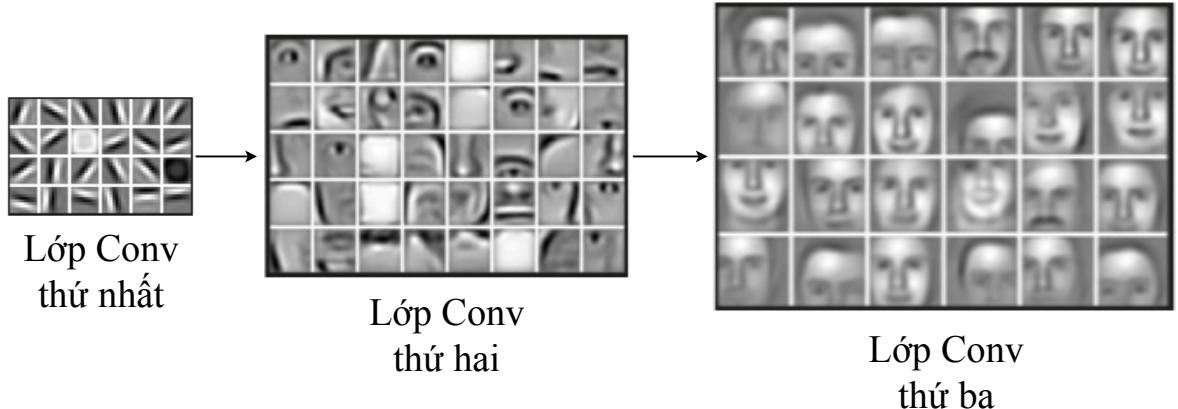
Dữ liệu đầu vào của CNN thông thường là ảnh được thể hiện dưới dạng một ma trận với kích thước $W \times H \times d$, với W là chiều rộng, H là chiều cao, d là số kênh của ảnh. Ví dụ: Ảnh RGB có 3 kênh màu thì $d = 3$, ảnh xám có 1 kênh màu thì $d = 1$.

Một mô hình CNN gồm một dãy các lớp, mỗi lớp sẽ có các hàm tính toán khác nhau, dữ liệu đầu ra của lớp trước là dữ liệu đầu vào của lớp sau, các lớp trong mô hình CNN cơ bản gồm ba thành phần chính là lớp Convolutional (Conv), lớp Pooling (Pool) và lớp Fully-Connected (FC). Dưới đây là hình minh họa một mô hình CNN gồm ba lớp Conv, ba lớp Pool và hai lớp FC.



Hình 2.8: Minh họa mô hình CNN

Hình dưới đây minh họa những chi tiết mà các lớp Conv tìm kiếm trên dữ liệu mặt người.



Hình 2.9: Minh họa ý nghĩa của các lớp Convolutional trong mô hình CNN (Nguồn từ [1])

Lớp Conv đầu sẽ tìm các chi tiết như góc cạnh và màu sắc. Lớp Conv tiếp theo sẽ kết hợp các chi tiết từ lớp đầu tiên để tìm các chi tiết trừu tượng hơn như mắt, mũi, môi, tai,... Lớp Conv thứ ba sử dụng các chi tiết trước đó để tìm những hình ảnh gương mặt khác nhau.

2.3.3.2 Lớp Convolutional

Lớp Convolutional (gọi tắt là Conv) là lớp chứa các ma trận tích chập (còn được gọi là Filter hoặc Kernel), mỗi ma trận tích chập có $l \times l \times d$ giá trị trọng số w và một tham số b có thể được thay đổi trong quá trình huấn luyện. Trong đó l là chiều dài cạnh của ma trận tích chập, d là số kênh của ma trận đầu vào. Các ma trận này sẽ được duyệt theo dạng cửa sổ trượt (Sliding Windows) với khoảng cách dịch chuyển (Stride) là S và tính tích chập với dữ liệu đầu vào, sau đó cộng với biến b . Để đảm bảo phép tích chập được thực hiện đầy đủ trên toàn ảnh thì Zero-padding được sử dụng, Zero-padding là việc thêm một khung có độ dày P gồm các giá trị 0 vào xung quanh ma trận đầu vào. Nếu kích thước ma trận đầu vào là $W \times H \times d$, thì kích thước ma trận đầu ra của lớp Conv là $W_o \times H_o \times d_o$. Kích thước ma trận đầu ra được tính bằng các công thức sau:

$$W_o = \frac{W - l + 2P}{S} + 1 \quad (2.5a)$$

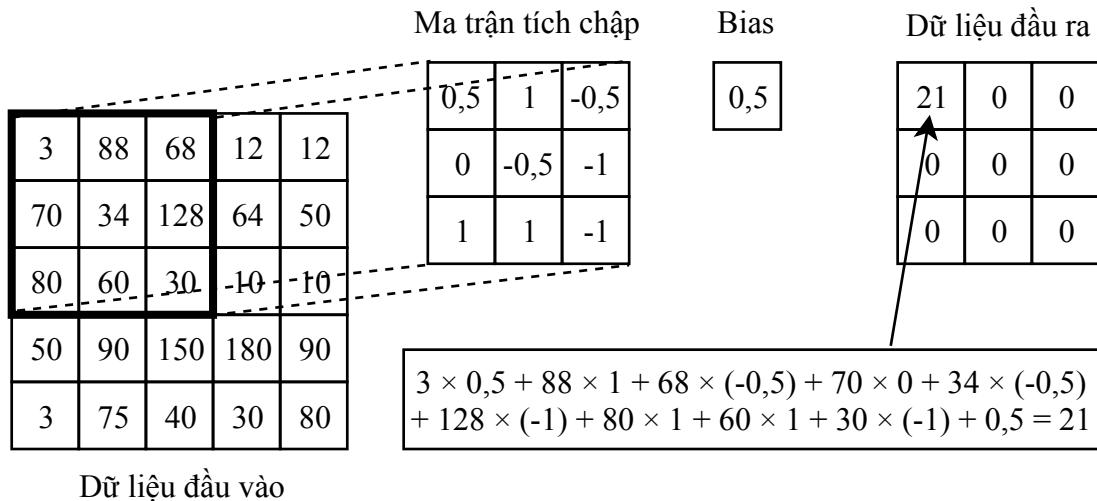
$$H_o = \frac{H - l + 2P}{S} + 1 \quad (2.5b)$$

$$d_o = N \quad (2.5c)$$

Trong đó l là Kích thước ma trận tích chập, khoảng cách dịch chuyển là S , Zero-padding có độ dày là P và N là số lượng ma trận tích chập.

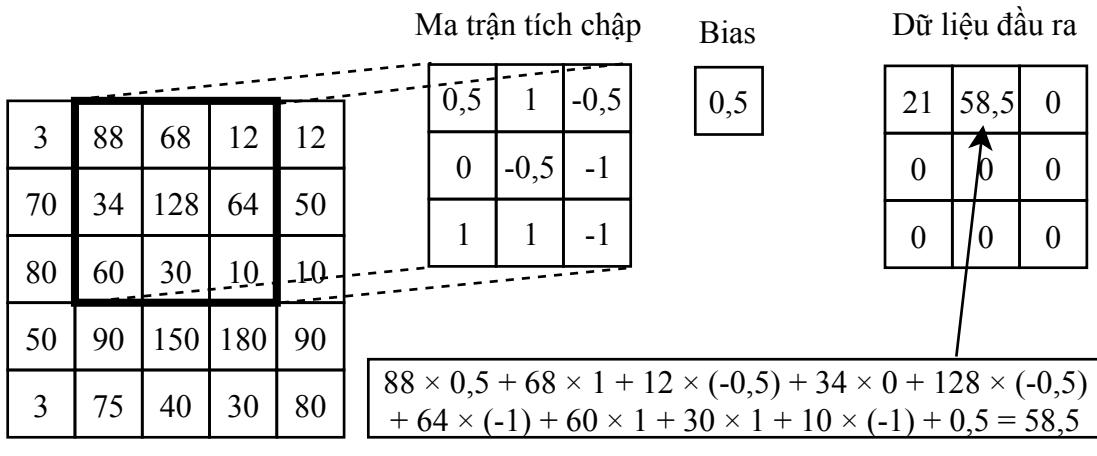
Dưới đây là một ví dụ minh họa một lớp Conv có dữ liệu đầu vào là một ma trận có kích thước $5 \times 5 \times 1$, kích thước ma trận tích chập là 3, số lượng ma trận tích chập của lớp là 1, khoảng cách dịch chuyển của ma trận là 1, Zero-padding có độ dày là 1, tham số b của ma trận tích chập có giá trị là 0,5.

Hình dưới đây mô tả cách tính tích chập với dữ liệu ở vị trí đầu tiên và cộng với tham số Bias để có được giá trị đầu ra thứ nhất.



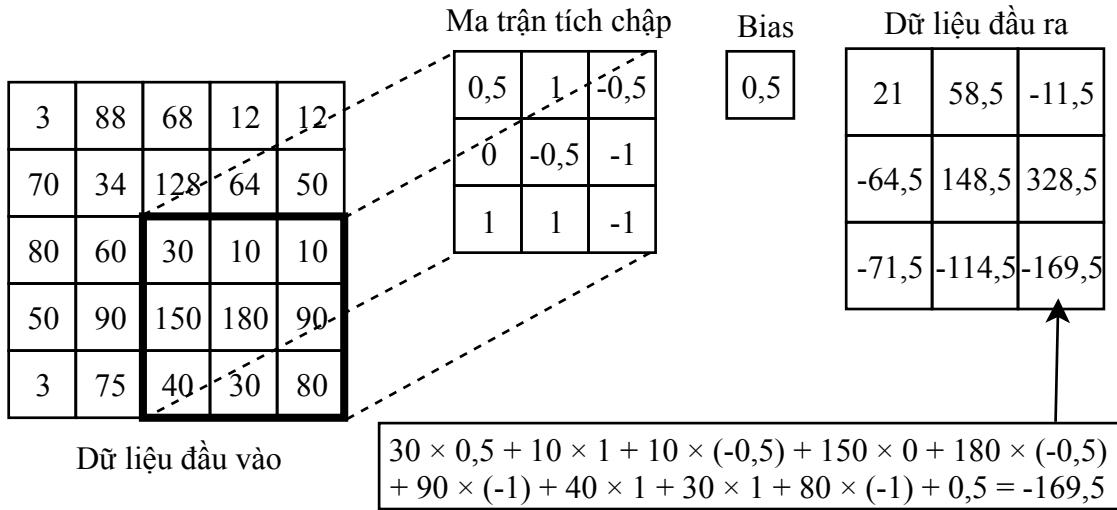
Hình 2.10: Minh họa cách hoạt động của lớp Convolutional (1)

Hình dưới đây mô tả cách tính giá trị đầu ra thứ hai.



Hình 2.11: Minh họa cách hoạt động của lớp Convolutional (2)

Quá trình tính toán được thực hiện tương tự cho đến khi duyệt xong dữ liệu đầu vào theo dạng cửa sổ trượt.



Hình 2.12: Minh họa cách hoạt động của lớp Convolutional (3)

2.3.3.3 Lớp Max Pooling

Lớp Max Pooling có chức năng làm giảm kích thước của ma trận đầu vào nhằm làm giảm độ phức tạp tính toán và số lượng tham số cần phải huấn luyện. Lớp Max Pooling hoạt động bằng cách dịch chuyển một cửa sổ trượt có kích thước $l_s \times l_s$ với khoảng cách dịch chuyển là S_p , ở mỗi vị trí giá trị lớn nhất trong cửa sổ trượt của mỗi kênh sẽ được giữ lại. Nếu cửa sổ trượt nằm ngoài ma trận dữ liệu đầu vào thì ma trận đầu vào sẽ được thêm các giá trị trừ vô cực ($-\infty$) ở biên. Lớp Max Pooling thường được cài đặt với kích thước của sổ trượt là 2×2 và khoảng cách dịch chuyển là 2.

Nếu kích thước ma trận đầu vào là $W \times H \times d$, thì kích thước của ma trận đầu ra $W_o \times H_o \times d_o$ của lớp Max Pooling được tính bằng các công thức sau:

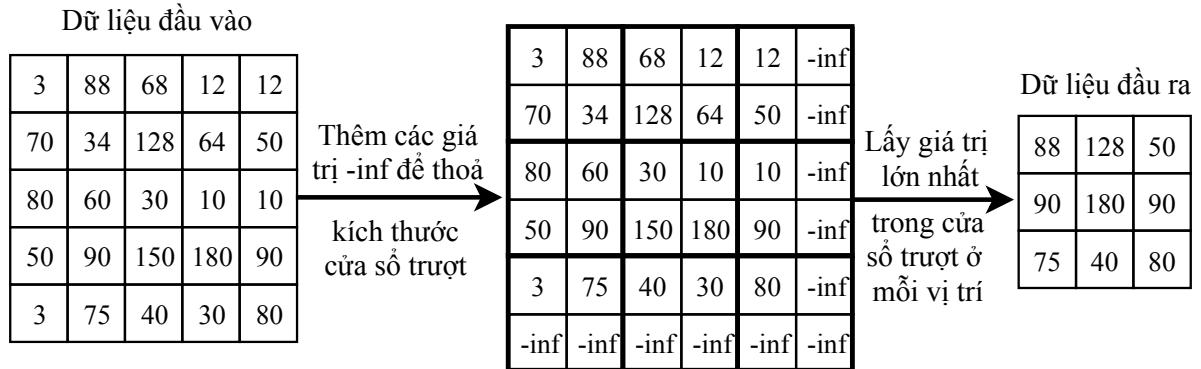
$$W_o = \frac{W - l_s}{S_p} + 1 \quad (2.6a)$$

$$H_o = \frac{H - l_s}{S_p} + 1 \quad (2.6b)$$

$$d_o = d \quad (2.6c)$$

Trong đó l_s là kích thước của cửa sổ trượt và khoảng cách dịch chuyển là S_p .

Dưới đây là một ví dụ minh họa một lớp Max Pooling có dữ liệu đầu vào là một ma trận có kích thước $5 \times 5 \times 1$, kích thước của cửa sổ trượt là 2 và khoảng cách dịch chuyển là 2.



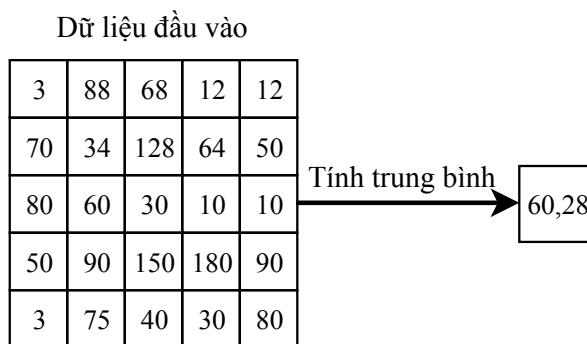
Hình 2.13: Minh họa cách hoạt động của lớp Max Pooling

2.3.3.4 Lớp Global Average Pooling

Cũng giống như Max Pooling, lớp Global Average Pooling có chức năng làm giảm kích thước của ma trận đầu vào nhằm làm giảm độ phức tạp tính toán và số lượng tham số cần phải huấn luyện. Lớp Global Average Pooling hoạt động bằng cách lấy giá trị trung bình của mỗi kênh trong ma trận đầu vào.

Nếu kích thước ma trận đầu vào là $W \times H \times d$, thì kích thước ma trận đầu ra của lớp Global Average Pooling là d .

Dưới đây là ví dụ minh họa cách hoạt động của lớp Global Average Pooling với dữ liệu đầu vào là một ma trận có kích thước $5 \times 5 \times 1$.



Hình 2.14: Minh họa cách hoạt động của lớp Global Average Pooling

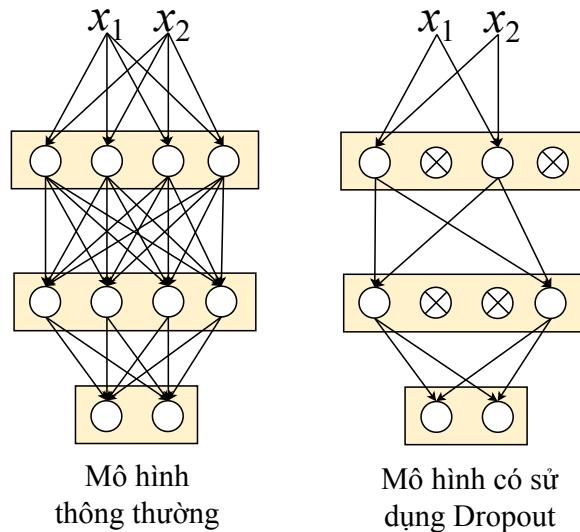
2.3.3.5 Lớp Dropout

Lớp Dropout là một lớp được sử dụng rất phổ biến và cho thấy được hiệu quả trong việc huấn luyện các mô hình học sâu. Lớp Dropout hoạt động bằng cách loại

bỏ (nhân với 0) ngẫu nhiên $p\%$ số lượng các giá trị trong ma trận đầu vào. Vì vậy, khi sử dụng thuật toán lan truyền ngược thì chỉ một phần các tham số của mô hình được cập nhật sau mỗi bước.

Lớp Dropout hoạt động hiệu quả là vì khi đánh giá độ chính xác của mô hình thì lớp Dropout không thực hiện loại bỏ mà sử dụng toàn bộ các giá trị đầu vào, điều này giống như việc sử dụng giá trị dự đoán trung bình của vô số mô hình nhỏ.

Dưới đây là hình minh họa mô hình có sử dụng và không sử dụng Dropout.



Hình 2.15: Minh họa mô hình có sử dụng và không sử dụng Dropout

Chương 3

Phương pháp đề xuất

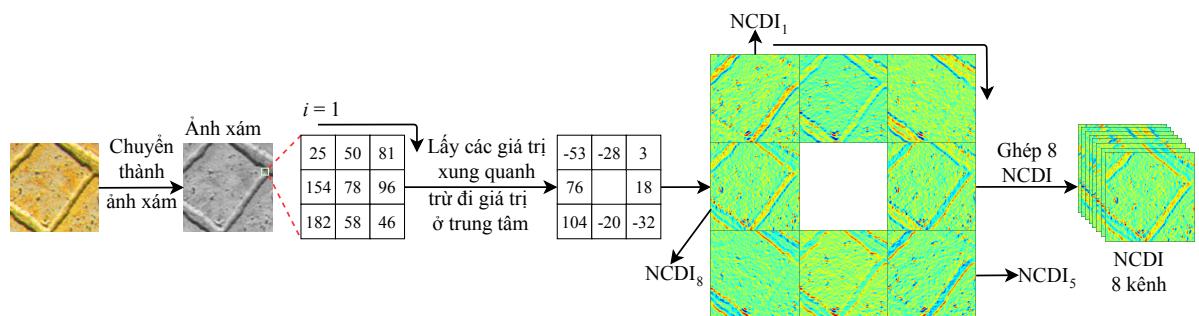
3.1 Tiền xử lý dữ liệu

3.1.1 Đặc trưng Neighbor-Center Difference Image (NCDI)

Như đã được giới thiệu ở phần 1.2.1, LBP đã làm mất đi thông tin về độ chênh lệch giữa các điểm ảnh do sử dụng hàm phân ngưỡng 1.2. Để khắc phục vấn đề này, Bing-Fei Wu và Chun-Hsien Lin đã đề xuất đặc trưng NCDI và mang lại hiệu quả trong bài toán nhận dạng biểu cảm của khuôn mặt [8]. Đặc trưng này được rút trích từ ảnh xám bằng việc di chuyển qua từ điểm ảnh, ở mỗi điểm ảnh thì P giá trị của NCDI tại vị trí (x, y) được tính bằng cách lấy các giá trị của P điểm ảnh xung quanh $\{g_i\}_{i=0}^{P-1}$ trừ đi giá trị của điểm ảnh trung tâm g_c bằng công thức:

$$\text{NCDI}(x, y)_i = g_i - g_c \quad (3.1)$$

Sau đó P ma trận NCDI sẽ được ghép lại thành một ảnh có P kênh, mỗi kênh sẽ có thông tin về góc cạnh (độ chênh lệch) theo một hướng. Dưới đây là hình minh họa quá trình trích xuất đặc trưng NCDI 8 kênh.



Hình 3.1: Minh họa quá trình trích xuất đặc trưng NCDI 8 kênh

3.1.2 Đặc trưng NCDI histogram và LBP NCDI

Thông tin về góc cạnh từ NCDI rất hữu ích trong nhiều ứng dụng về xử lý ảnh. Do đó, hai loại đặc trưng histogram được rút trích từ NCDI để áp dụng cho bài toán texture classification:

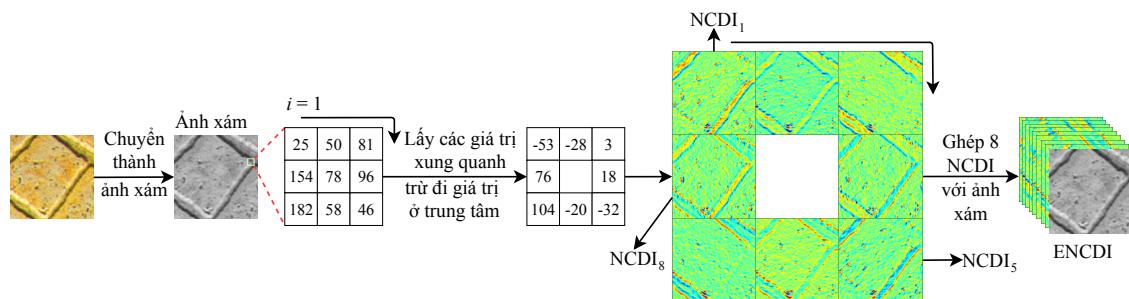
- Loại đặc trưng histogram đầu tiên được trích xuất bằng cách thống kê tầng suất xuất hiện của các giá trị từ -255 đến 255 của từng kênh NCDI. Đặc trưng histogram trên NCDI sẽ có thông tin về tầng suất của độ chênh lệch giữa các điểm ảnh.
- Để trích xuất loại đặc trưng histogram thứ hai. Đầu tiên, thuật toán LBP sẽ được áp dụng trên mỗi kênh của NCDI. Sau đó, đặc trưng histogram thứ hai được trích xuất bằng cách thống kê tầng suất xuất hiện của 256 giá trị LBP. Đặc trưng LBP NCDI sẽ có thông tin về sự tương quan độ khác nhau giữa các điểm ảnh.

Do hai đặc trưng NCDI histogram và LBP NCDI có thông tin bổ trợ cho nhau nên nhóm quyết định kết hợp hai đặc trưng để áp dụng cho bài toán texture classification.

3.1.3 Đặc trưng Enhanced NCDI (ENCDI)

Do đặc trưng NCDI chỉ chứa thông tin về các góc cạnh nhưng không chứa thông tin về độ sáng của ảnh. Khoa luận đã đề xuất một đặc trưng mới và đặt tên là Enhanced NCDI. Đặc trưng mới khác với NCDI là ghép thêm ảnh xám vào để có thêm thông tin về độ sáng của ảnh.

Dưới đây là hình ảnh minh họa các bước trích xuất đặc trưng ENCDI. Các bước rút trích đặc trưng NCDI 8 kênh tương tự như đã được trình bày ở trên, điểm khác biệt của ENCDI so với NCDI là ở bước ghép thêm ảnh xám.

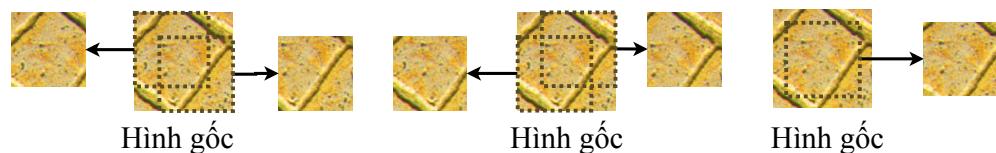


Hình 3.2: Minh họa quá trình trích xuất đặc trưng ENCDI

3.1.4 Kỹ thuật Multi-crop

Kỹ thuật Multi-crop thường được sử dụng để tăng cường thêm dữ liệu huấn luyện. Kỹ thuật này hoạt động bằng cách cắt một số ảnh nhỏ trong mỗi ảnh, để quyết định một ảnh gốc thuộc lớp nào thì kết quả dự đoán của mô hình trên các ảnh nhỏ của ảnh gốc đó sẽ được lấy trung bình.

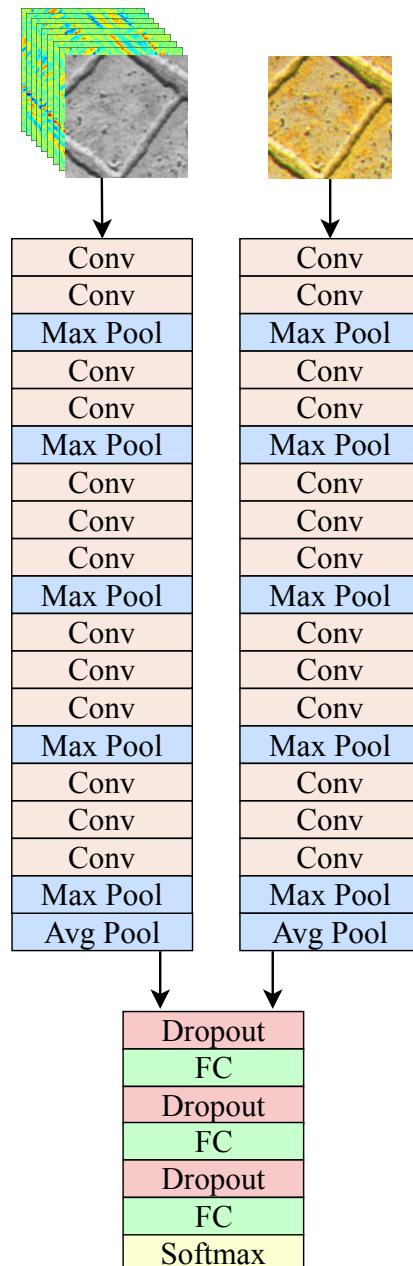
Trong bốn bộ dữ liệu được dùng để kiểm tra thì kỹ thuật Multi-crop được sử dụng trên ba bộ gồm Outex-TC00013, USPTex và STex. Trên mỗi hình của ba bộ dữ liệu này, năm ảnh nhỏ với kích thước 96×96 được cắt từ mỗi ảnh, trong đó bốn ảnh nhỏ được cắt từ bốn góc và một ảnh nhỏ được cắt ở giữa ảnh gốc. Đối với bộ dữ liệu còn lại là New BarkTex, do độ phân giải thấp nên kỹ thuật Multi-crop không được sử dụng. Hình dưới đây minh họa cách cắt 5 hình nhỏ có kích thước 96×96 từ một hình có kích thước 128×128 .



Hình 3.3: Minh họa cách cắt 5 hình nhỏ từ một hình gốc

3.2 Mô hình đề xuất

Mô hình học sâu mà khoá luận đề xuất để giải quyết bài toán Texture Classification là một mô hình hai luồng được tạo thành từ hai mô hình VGG-16 [9], mô hình này có hai dữ liệu đầu vào là ảnh RGB và ENCDI. Một lớp Global Average Pooling sẽ được thêm vào sau lớp Max-pooling cuối cùng của mỗi luồng để giảm đi vấn đề overfitting, sau đó là ba lớp FC và một lớp Softmax. Trước mỗi lớp FC có một lớp Dropout với tỉ lệ loại bỏ là 50%.



Hình 3.4: Hình ảnh minh họa mô hình được đề xuất.

Chương 4

Quá trình thực nghiệm

4.1 Các tập dữ liệu

Phương pháp đề xuất của khoá luận được đánh giá trên bốn bộ dữ liệu texture màu bao gồm BarkTex [10], Outex-TC-00013 [2], USPTex [11] và STex.

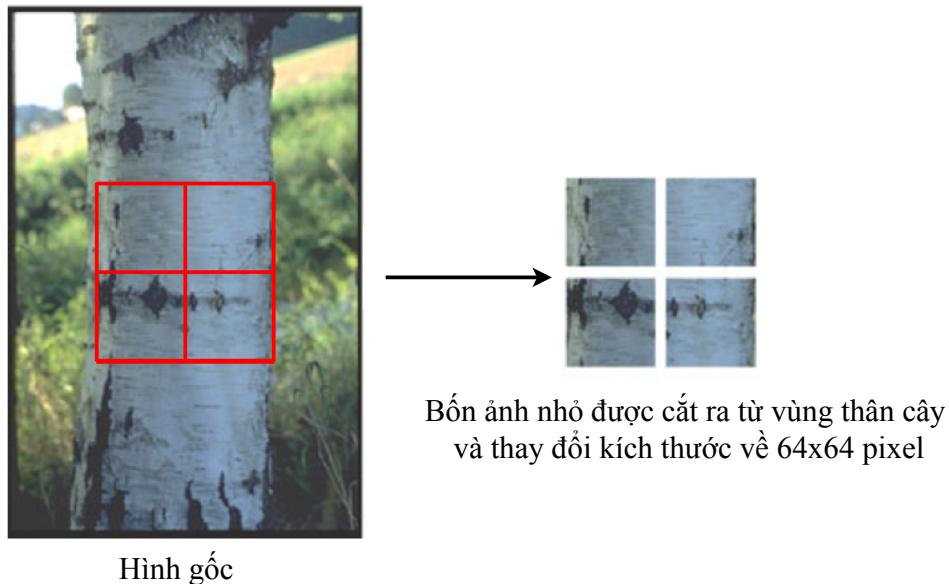
Bảng 4.1: Bảng tóm tắt thông tin của bốn tập dữ liệu được sử dụng trong quá trình thực nghiệm.

Tên tập dữ liệu	Kích thước ảnh	Số lớp	Số ảnh huấn luyện	Số ảnh kiểm tra	Tổng cộng
New BarkTex	64×64	6	816	816	1632
Outex-TC-00013	128×128	68	680	680	1360
USPTex	128×128	191	1146	1146	2292
STex	128×128	476	3808	3808	7616

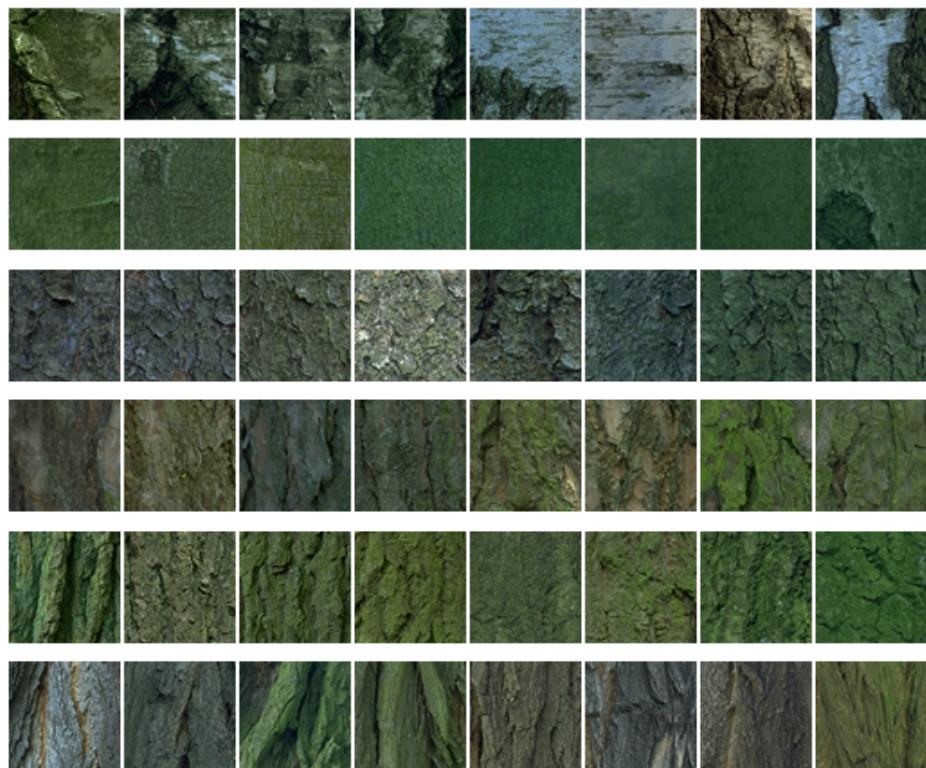
4.1.1 New-BarkTex

Bộ dữ liệu New BarkTex gồm sáu loại vỏ cây, mỗi loại vỏ cây có 68 hình. Để xây dựng bộ dữ liệu này, một vùng có kích thước 128×128 pixel được chọn ra trong mỗi hình của mỗi loại vỏ cây. Sau đó vùng được chọn sẽ được chia ra thành 4 hình nhỏ với kích thước 64×64 pixel, do đó mỗi lớp có $4 \times 68 = 272$ hình. Để đảm bảo tập ảnh dùng để huấn luyện và tập ảnh dùng để kiểm tra có độ tương quan ít nhất có thể, 4 hình nhỏ (64×64 pixel) được cắt ra từ một ảnh gốc sẽ được cho vào cùng tập huấn luyện hoặc tập kiểm tra.

Dưới đây là một số ảnh mẫu trong bộ dữ liệu New BarkTex. Các ảnh của cùng một loại cây được xếp cùng một hàng.



Hình 4.1: Cách lấy ảnh của bộ dữ liệu New BarkTex



Hình 4.2: Một số ảnh mẫu trong bộ dữ liệu New BarkTex

4.1.2 Outex-TC00013

Bộ dữ liệu Outex-TC00013 gồm 68 loại texture khác nhau như vải, giấy, gỗ, gạch,... Tất cả ảnh được chụp dưới điều kiện ánh sáng giống nhau, mỗi loại vật liệu có một ảnh với kích thước 746×538 pixel. Mỗi ảnh sau đó được cắt thành 20 ảnh với kích thước 128×128 pixel và không trùng lặp nhau, do đó cả bộ dữ liệu có $68 \times 20 = 1360$

hình. Để tạo ra tập ảnh huấn luyện và tập ảnh kiểm tra người ta chia 20 hình nhỏ (128×128 pixel).

Dưới đây là 68 hình, mỗi hình được lấy từ một lớp trong bộ dữ liệu Outex-TC00013.

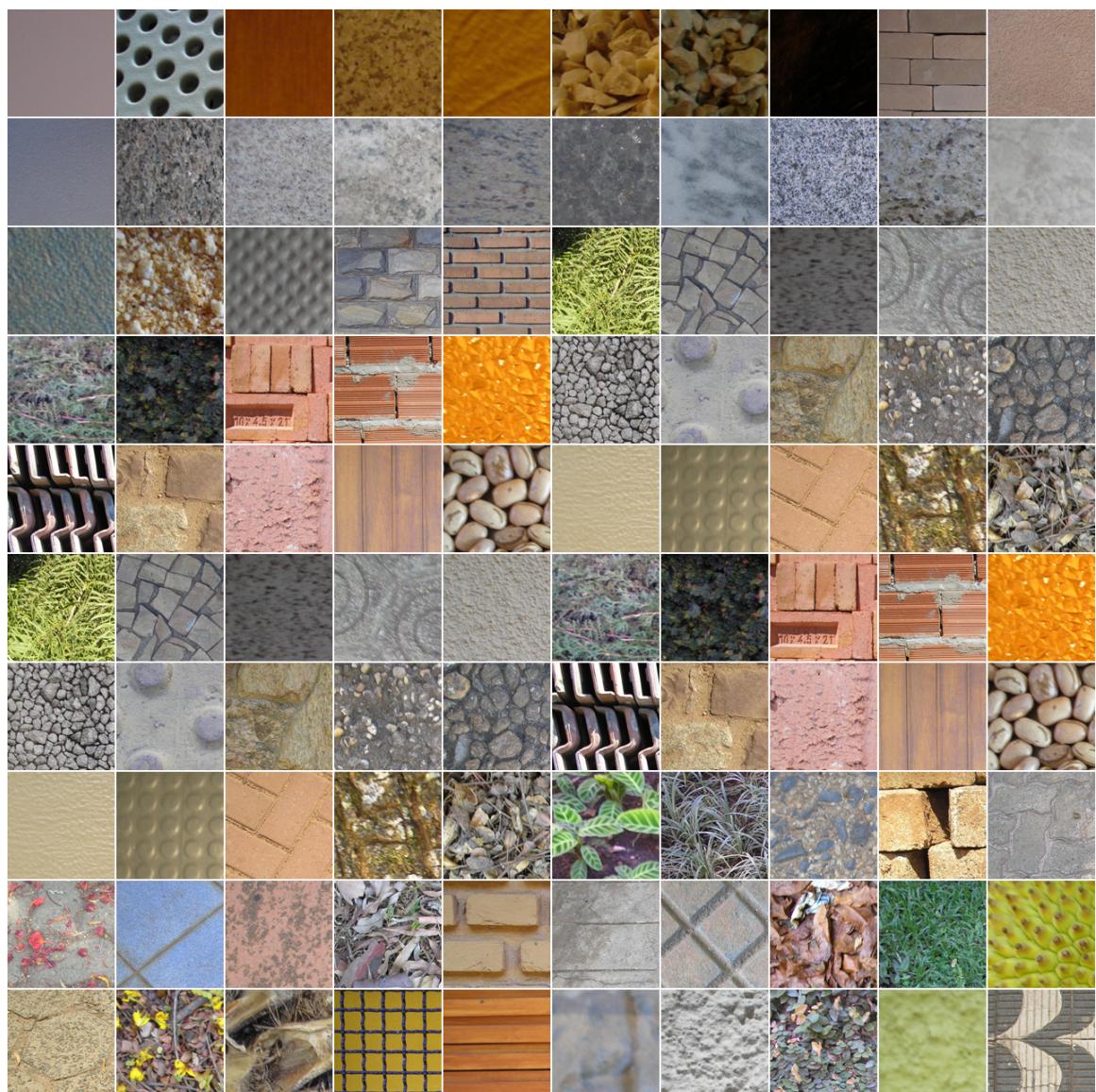


Hình 4.3: Một số ảnh mẫu trong bộ dữ liệu Outex-TC00013

4.1.3 USPTex

Bộ dữ liệu USPTex có tổng cộng 191 loại texture khác nhau được chụp dưới cùng điều kiện ánh sáng. Trong bộ dữ liệu gồm những loại texture như gạo, đậu, gạch, lá cây,... Mỗi texture có một ảnh với kích thước 512×384 pixel. Mỗi ảnh sau đó được cắt thành 12 ảnh không trùng lặp nhau, những ảnh này có cùng kích thước là 128×128 pixel, do đó cả bộ dữ liệu có $191 \times 12 = 2292$ hình. Để tạo ra tập ảnh huấn luyện và tập ảnh kiểm tra người ta chia 12 hình nhỏ (128×128 pixel).

Dưới đây là 100 hình, mỗi hình được lấy từ một lớp trong 191 lớp của bộ dữ liệu USPTex.

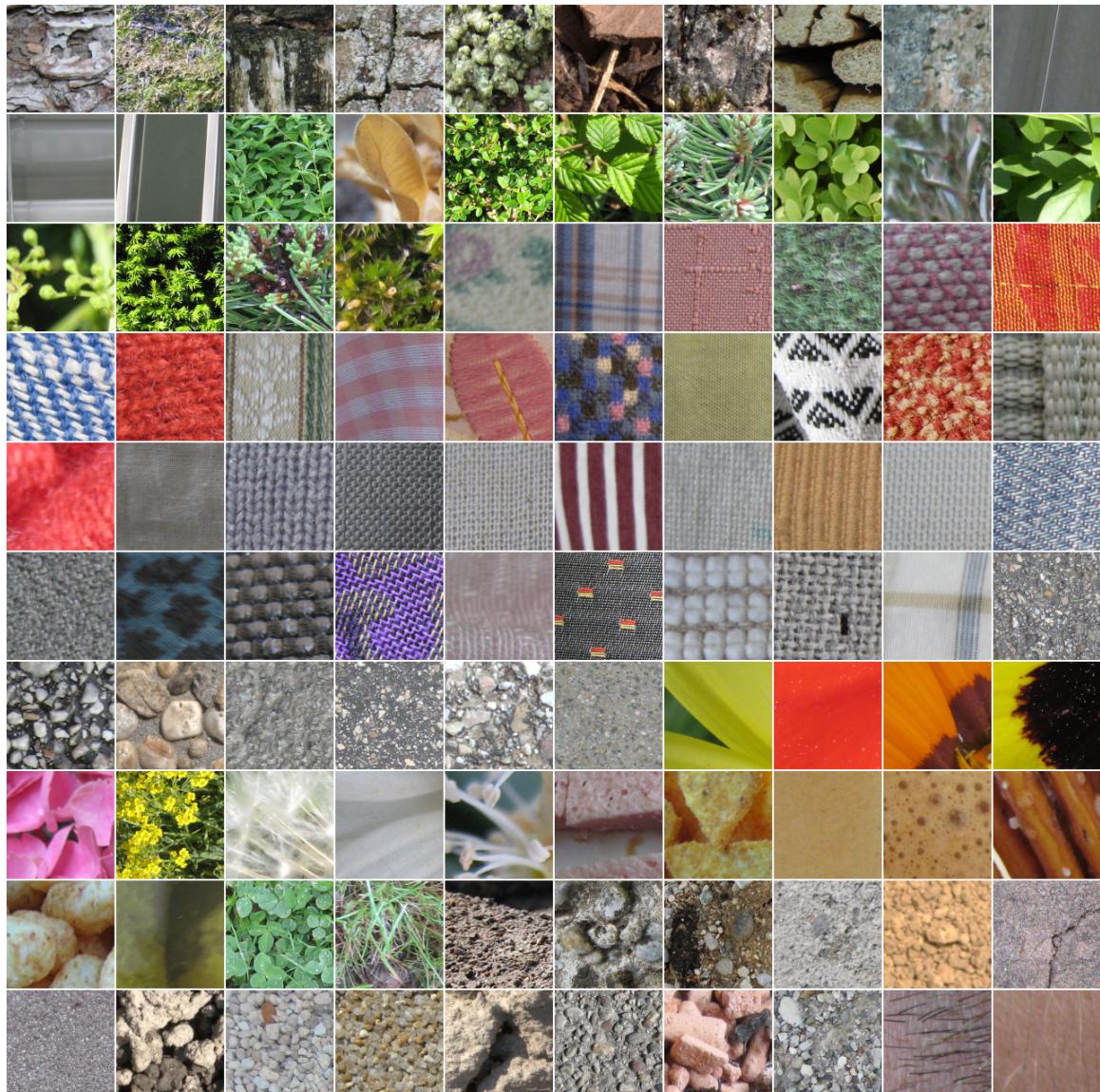


Hình 4.4: Một số ảnh mẫu trong bộ dữ liệu USPTex

4.1.4 STex

Bộ dữ liệu STex chứa 476 hình với kích thước 512×512 pixel, mỗi hình này được cắt thành 16 hình với kích thước 128×128 pixel. Do đó, bộ dữ liệu có tổng cộng $476 \times 16 = 7616$ hình nhỏ với kích thước 128×128 pixel.

Dưới đây là 100 hình, mỗi hình được lấy từ một lớp trong 476 lớp của bộ dữ liệu USPTex.



Hình 4.5: Một số ảnh mẫu trong bộ dữ liệu STex

4.2 Thiết lập cho các thí nghiệm

4.2.1 Thiết lập cho thí nghiệm đánh giá đặc trưng NCDI histogram và LBP NCDI

Kết quả của đặc trưng NCDI histogram và LBP NCDI được đánh giá và so sánh với đặc trưng LBP và một số biến thể khác của LBP. Tất cả thí nghiệm được thực hiện trên Matlab 2015b và thực thi trên một máy tính của Google Cloud Platform với cấu hình 8 CPU 2,5 GHz, 30 GB RAM. Tất cả thí nghiệm được thực hiện trên cùng tập huấn luyện và tập kiểm tra của bốn bộ dữ liệu để kết quả là công bằng nhất. Các đặc trưng được rút trích từ từng kênh của mỗi ảnh RGB. Sau đó, đặc trưng từ ba kênh của mỗi ảnh sẽ được nối lại thành dãy đặc trưng đại diện cho ảnh đó. Sau khi các đặc trưng được rút trích thì thuật toán KNN (với K bằng 1) được sử dụng để phân lớp và đánh giá độ chính xác của mỗi loại đặc trưng.

4.2.2 Thiết lập cho thí nghiệm đánh giá mô hình học sâu kết hợp đặc trưng cục bộ

Các mô hình học sâu được cài đặt bằng framework Keras trên Python. Quá trình huấn luyện và kiểm tra các mô hình học sâu được thực thi trên một máy của Google Cloud Platform với cấu hình 8 CPUs 2.5 GHz, 52 GB RAM, 1 NVIDIA Tesla P100 GPU 16 GB RAM.

Kỹ thuật multi-crop được sử dụng để tăng cường dữ liệu huấn luyện cho các mô hình học sâu. Năm vùng của mỗi ảnh sẽ được cắt như sau: bốn vùng được cắt từ bốn gốc của mỗi ảnh và một vùng được cắt từ trung tâm của ảnh. Trong quá trình huấn luyện, 5 vùng từ mỗi ảnh của tập ảnh huấn luyện sẽ được sử dụng làm dữ liệu huấn luyện. Trong quá trình kiểm tra, kết quả của mỗi ảnh kiểm tra sẽ được tính bằng cách lấy trung bình kết quả phân phối xác suất của 5 vùng được cắt từ ảnh đó. Sau đó lớp của ảnh được quyết định là lớp có kết quả phân phối xác suất trung bình cao nhất. Kỹ thuật multi-crop không được sử dụng cho bộ dữ liệu New BarkTex vì kích thước ảnh quá nhỏ. Đối với ba bộ dữ liệu còn lại, năm vùng với kích thước 96×96 pixel sẽ được cắt ra từ mỗi ảnh.

Để kết quả so sánh được công bằng, tất cả mô hình học sâu được huấn luyện với cùng những tham số của mô hình: Số lượng ảnh cho mỗi lần huấn luyện là 16 đối với bộ dữ liệu New BarkTex và 64 đối với ba bộ dữ liệu còn lại. Tất cả mô hình dùng thuật toán tối ưu Adam [12]. Tốc độ học ban đầu của các mô hình là 10^{-5} , sau đó tốc độ học được giảm xuống 10^{-6} khi độ chính xác trên tập kiểm tra ngừng tăng lần thứ nhất. Khi độ chính xác ngừng tăng lần thứ hai, tốc độ học được giữ nguyên là 10^{-6} và số lượng ảnh cho mỗi lần huấn luyện được tăng lên 256.

Để đánh giá kết quả của mô hình VGG-16 RGB, mô hình đã được huấn luyện trên ImageNet được sử dụng. Ba lớp FC của mô hình được thay thế bằng ba lớp FC mới, hai lớp dropout được thêm vào giữa ba lớp FC với tỉ lệ loại bỏ là 50%. Đối với mô hình học sâu kết hợp đặc trưng cục bộ VGG-16 RGB & ENCDI, kiến trúc mô hình được xây dựng như đã trình bày ở phần 3.2.

4.3 Kết quả

4.3.1 Kết quả đánh giá đặc trưng NCDI histogram và LBP NCDI

Trong phần này, kết quả của đặc trưng NCDI histogram và LBP NCDI sẽ được so sánh với LBP và một số biến thể của LBP.

Bảng 4.2: Bảng thống kê độ chính xác (tính bằng %) của LBP và các phương pháp đề xuất trên bốn bộ dữ liệu gồm New-BarkTex, Outex-TC00013, USPTex và STEX.

	New BarkTex	Outex-TC-00013	USPTex	STex
LBP RGB	76,6	86,0	85,3	85,5
LBP NCDI	74,4	88,9	82,3	87,8
NCDI Histogram	69,4	86,3	80,8	76,9
NCDI Histogram & LBP NCDI	78,3	90,2	88,7	89,1

Từ bảng 4.2, có thể thấy rằng phương pháp đề xuất kết hợp hai loại đặc trưng NCDI histogram và LBP NCDI cho kết quả tốt hơn so với LBP thông thường. So với LBP thông thường, phương pháp đề xuất đã đạt được kết quả tốt hơn trên bốn bộ dữ liệu New BarkTex, Outex-TC-00013, USPTex, và STEX với khoảng cách lần lượt là 1,7%, 4%, 3,4% và 3,5%.

Từ bảng 4.3, có thể thấy rằng phương pháp kết hợp đặc trưng NCDI histogram và LBP NCDI đạt được kết quả tốt hơn nhiều phương pháp khác dựa vào kết quả thực nghiệm trên bốn bộ dữ liệu. Trên bộ dữ liệu New BarkTex, USPTex, STEX thì phương pháp đề xuất đạt được kết quả tốt hơn tất cả các phương pháp khác với khoảng cách trên 0,6%, 0,3%, 1,5%. Trên bộ dữ liệu Outex-TC-00013, Trên bộ dữ liệu STex, phương pháp đề xuất đạt được kết quả tốt hơn tất cả các phương pháp khác với khoảng cách trên 1,5%. Trên bộ dữ liệu USPTex, độ chính xác của phương pháp đề xuất tốt hơn một khoảng trên 0,3%. Trên bộ dữ liệu New BarkTex, LTP cho kết quả tốt hơn phương pháp đề xuất một khoảng 0,4%, tuy nhiên phương pháp

Bảng 4.3: Bảng thống kê độ chính xác (tính bằng %) của phương pháp đề xuất và một số phương pháp khác trên bốn bộ dữ liệu New BarkTex, Outex-TC-00013, USPTex, và STex.

Methods	New BarkTex	Outex-TC-00013	USPTex	STex
Color angles LBP [13]	71,0	86,2	79,1	-
Wavelet coefficients [14]	-	89,7	-	77,6
Color contrast occurrence matrix [15]	-	82,6	-	76,7
Soft color descriptors [16]	-	81,4	58,0	55,3
LBP and local color contrast [17]	71,0	85,3	82,9	-
CLBP [6]	72,8	84,4	72,3	-
Mix color order LBP histogram [18]	77,7	87,1	84,2	-
LTP [4]	76,1	90,6	88,4	87,0
LPQ [19]	66,2	81,4	86,6	87,6
TPLBP [20]	61,3	75,0	80,0	71,7
LBP Median [21]	72,3	83,0	84,1	81,9
NCDI Histogram & LBP NCDI	78,3	90,2	88,7	89,1

đề xuất đạt kết quả cao hơn LTP một khoảng 2,2% trên New BarkTex, 0.3% trên USPTex, và 2.1% trên STex.

4.3.2 Kết quả đánh giá mô hình học sâu kết hợp đặc trưng cục bộ

Trong phần này, kết quả của mô hình đề xuất sẽ được so sánh với mô hình VGG-16 thông thường và những kết quả cao nhất hiện tại trên bốn bộ dữ liệu gồm New-BarkTex, Outex-TC00013, USPTex và STex. Để đánh giá độ chính xác của các mô hình, các mô hình được huấn luyện với cùng tập dữ liệu huấn luyện và thực hiện đánh giá độ chính xác dựa trên cùng tập dữ liệu kiểm tra. Tham số của các mô hình

được thiết lập giống nhau để kết quả của các mô hình được đánh giá một cách công bằng nhất.

Bảng 4.4: Bảng thống kê các thuật toán có độ chính xác (tính bằng %) cao nhất hiện tại và kết quả của mô hình đề xuất trên bốn bộ dữ liệu gồm New-BarkTex, Outex-TC00013, USPTex và STex.

Phương pháp	New BarkTex	Outex-TC-00013	USPTex	STex
Wavelet coefficients [14]	-	89,7	-	77,6
Color contrast occurrence matrix [15]	-	82,6	-	76,7
Combine color and LBP-based features [22]	-	90,2	95,7	-
Quaternion-Michelson Descriptor [23]	-	91,3	94,2	-
Halftoning Local Derivative Pattern and Color Histogram [24]	-	88,2	93,9	-
3D Color histogram [25]	79,9	94,7	-	-
Soft color descriptors [16]	-	81,4	58	55,3
LBP and local color contrast [17]	71	85,3	82,9	-
CLBP [6]	72,8	84,4	72,3	-
Mix color order LBP histogram [18]	77,7	87,1	84,2	-
Color angles [26]	80,2	86,2	88,8	-
Sparse score [27]	81,3	93,4	93,2	-
DRLBP [28]	61,4	89	89,4	89,4
MCSBS [29]	87,8	92,9	97,3	96,7
ASL-based MCSHS [29]	86,8	95,3	97,6	96,1
ICS-based MCSHS [29]	92,6	95,6	97,2	94,1
VGG-16 RGB	94,4	93,5	99,3	96,2
VGG-16 RGB & ENCDI (Mô hình đề xuất)	95,6	94,7	99,6	97,6

Từ bảng 4.4, ta có thể thấy rằng mô hình đề xuất luôn tốt hơn mô hình VGG-16 thông thường nhờ kết hợp thêm thông tin từ đặc trưng ENCDI. So với các kết quả tốt nhất trên bốn bộ dữ liệu thì mô hình đề xuất có kết quả tốt hơn trong ba bộ dữ liệu là New-BarkTex, USPTex và STex.

Trên bộ dữ liệu New-BarkTex, vì độ phân giải thấp nên kỹ thuật Multi-crop không được sử dụng. Tuy nhiên các mô hình học sâu đều cho kết quả tốt hơn các hướng giải quyết khác với một khoảng cách tách biệt hơn 1,8%. Mô hình đề xuất đạt độ chính xác là 95,6%, tốt hơn 1,2% so với mô hình VGG-16 thông thường.

Dối với bộ dữ liệu Outex-TC00013, mô hình đề xuất không đạt kết quả cao hơn một số hướng giải quyết khác. Tuy nhiên, so với các kết quả tốt nhất trên ba bộ dữ liệu còn lại thì mô hình đề xuất đã đạt được kết cách biệt đáng kể là 3% trên New-BarkTex, 2% trên USPTex và 0,9% trên STex.

Dựa vào kết quả thực nghiệm trên bộ dữ liệu USPTex, các mô hình học sâu đã đạt được kết quả tốt hơn 1,7% so với kết quả tốt nhất hiện tại. Mô hình đề xuất đạt được độ chính xác 99,6% và mô hình VGG-16 thông thường đạt 99,3%.

Dối với bộ dữ liệu STex, mô hình đề xuất đạt độ chính xác là 97,6%, kết quả này cao hơn kết quả của mô hình VGG-16 thông thường 1,4% và hơn kết quả tốt nhất hiện tại trên bộ dữ liệu này một khoảng 0.9%.

Mô hình VGG-16 huấn luyện trên ảnh RGB kích thước 96×96 pixel có 50,653,060 tham số để huấn luyện. Mô hình kết hợp đặc trưng cục bộ có tổng cộng 50,691,140 tham số để huấn luyện. Trên bộ dữ liệu OuTex-TC-00013, thời gian để huấn luyện một lượt trên 3400 ảnh với kích thước 96×96 pixel là 6,3 giây đối với mô hình VGG-16 và 12,1 giây đối với mô hình kết hợp đặc trưng cục bộ VGG-16 RGB & ENCDI.

Chương 5

Kết luận

Thông qua quá trình nghiên cứu, nhóm đã đề xuất được hai loại đặc trưng mới và một hướng giải quyết mới cho bài toán Texture Classification.

Hai loại đặc trưng được đề xuất là NCDI histogram và LBP NCDI. Hai đặc trưng này rút trích thông tin về góc cạnh từ NCDI và hai đặc trưng này có thông tin bổ trợ cho nhau. Do đó, việc kết hợp hai đặc trưng được đề xuất để tăng độ chính xác. Thông qua quá trình thực nghiệm cho thấy phương pháp kết hợp đặc trưng NCDI histogram và LBP NCDI đạt được kết quả tốt hơn so với nhiều phương pháp khác trên bốn bộ dữ liệu. Hướng phát triển tiếp theo của việc sử dụng hai đặc trưng này là tìm cách giảm đi số lượng thông tin thừa và áp dụng đặc trưng này cho những bài toán khác liên quan đến thị giác máy tính.

Ở phương pháp kết hợp rút trích đặc trưng cục bộ và học sâu, nhóm đã đề xuất được một hướng giải quyết mới cho bài toán Texture Classification với độ chính xác cao hơn hướng giải quyết trước đó, hướng giải quyết này được phát triển dựa trên những công trình nghiên cứu trước đó như mô hình VGG-16 [9] và đặc trưng NCDI [8]. Khoa luận đã trình bày một phiên bản mở rộng của LBP, đặc trưng đề xuất có tên ENCDI được phát triển dựa trên NCDI của Bing-Fei Wu và Chun-Hsien Lin [8]. Đồng thời, khoa luận cũng đề xuất một mô hình học sâu gồm hai luồng để trích xuất dữ liệu từ ảnh RGB và ENCDI. Qua quá trình thực nghiệm, mô hình hai luồng có sử dụng thêm thông tin từ đặc trưng ENCDI luôn cho kết quả tốt hơn so với mô hình học sâu thông thường chỉ sử dụng ảnh RGB.

Qua quá trình thực nghiệm cho thấy mô hình đề xuất đã đạt độ chính xác cao hơn các thuật toán có kết quả tốt nhất hiện tại trên ba bộ dữ liệu gồm New-BarkTex, USPTex và STex với khoảng cách lần lượt là 3%, 2% và 0,9%. Trên bộ dữ liệu Outex-TC00013, mô hình đề xuất không đạt kết quả tốt hơn một số phương pháp khác, tuy nhiên trên ba bộ dữ liệu còn lại thì các phương pháp này có kết quả thấp hơn mô hình đề xuất một khoảng đáng kể. Mặc dù có kết quả cao hơn nhiều phương pháp khác nhưng hướng giải quyết của khoa luận có hạn chế về tài nguyên tính toán vì phải dùng mô hình hai luồng. Hướng phát triển tiếp theo của khoa luận là áp dụng các

phương thức rút trích đặc trưng khác để giải quyết bài toán Texture Classification, vì hướng giải quyết hiện vẫn chưa khai thác hết các biến thể khác của LBP và những phương thức rút trích đặc trưng khác.

Hai loại đặc trưng mới mà nhóm đề xuất hiện đang được xét duyệt tại hội nghị quốc tế Industrial Networks and Intelligent Systems (INISCOM). Dưới đây là bài báo của nhóm có tên "LBP-based edge information for color texture classification".

Phương pháp đề xuất của nhóm đã được chấp nhận tại hội nghị quốc tế Multimedia Analysis and Pattern Recognition (MAPR). Dưới đây là bài báo của nhóm có tên "Feeding Convolutional Neural Network by hand-crafted features based on Enhanced Neighbor-Center Different Image for color texture classification".

LBP-based edge information for color texture classification

Duc Phan Van Hoai and Vinh Truong Hoang

Faculty of Information Technology
Ho Chi Minh City Open University, Vietnam
e-mail: 1551010028duc@ou.edu.vn; vinh.th@ou.edu.vn

Abstract. In this paper, we propose to extract two types of feature from Neighbor-Center Difference Image (NCDI). NCDI is a variant of Local Binary Pattern (LBP) and originally used as input for Convolutional Neural Network (CNN). NCDI is a high dimensional feature and thus histograms are extracted from these NCDI features to mainly store useful information for statistical analysis. Two types of histograms are extracted from NCDI and then concatenated to further capture useful information. Experimental results on several benchmark color texture datasets show that the proposed approaches outperform the original LBP with a large margin in accuracy on several benchmark color texture datasets.

Keywords: LBP · neighbor-center different image · color texture classification

1 INTRODUCTION

Texture analysis is one of the most important fields of computer vision with a wide range of real-life applications, including industrial inspection, medical imaging, object detection, content-based image retrieval, and facial analysis. In reality, the texture of the same material or object varies in illumination, orientation, scale, and rotation. Therefore, it needs a robust descriptor to characterize and discriminate different classes. Various approaches have been proposed to overcome these drawbacks in illumination, orientation, and other visual appearance problems. Many researchers have been concentrated to propose the new efficiency descriptor. Most approaches are based on local and global techniques.

One simple yet efficient local descriptor is Local Binary Pattern (LBP) introduced by Ojala et al. [1]. LBP is a computational efficiency operator with high discriminative power and robustness against illumination. However, it may not work properly for noisy images due to its threshold function [?]. Various variants of LBP and its extension have been proposed to minimize its limitation [2]. Lu et al. [3] have proposed Neighbor-Center Difference Vector (NCDV) which is extracted by subtracting the center pixel value from its neighboring pixel values. They extract NCDV features of different sizes from several non-overlapped blocks of training samples. For NCDV features extracted from each block, they

train one projection to map it into a binary feature vector. Then, they cluster these binary codes into a codebook and encode these binary codes within the same face as a histogram feature vector. Finally, an age ranker is trained on these histograms. Their approach has shown to provide very good results on several datasets.

Recently, Wu and Lin [4] have proposed a new descriptor based on NCDV to capture edge information, namely Neighbor-Center Difference Image (NCDI). Normally, Convolutional Neural Network (CNN) model takes RGB images as input. However, Wu and Lin [4] have proposed to feed CNN model with hand-crafted feature NCDI which collects NCDV from all patches to reconstruct the image. This approach has shown to improve the accuracy in the facial expression recognition task. Edge information is useful to a wide range of computer vision tasks but NCDI is a high dimensional feature. Therefore, we propose to extract two types of histogram feature from NCDI to mainly store useful information and use it for texture recognition task.

The rest of this paper is organized as follows. Section 2 introduces the feature extracting methods. Next, four benchmark color texture datasets and experimental result of the proposed approaches are introduced in section 3. Finally, the conclusion is discussed in section 4.

2 PROPOSED APPROACH

LBP is a powerful local descriptor to deal with texture and related to classification tasks. LBP operator takes values of points on a circular neighborhood, thresholds the pixel values of the neighborhood at the value of the central pixel value. The binary results are then used to form an integer LBP code. The formula to compute the $\text{LBP}_{P,R}$ code from P circular neighbors of radius R is defined as:

$$\text{LBP}_{P,R} = \sum_{i=0}^{P-1} \theta(g_i - g_c) \times 2^i \quad (1)$$

where g_c is the value of central pixel and g_i is the value of i th neighborhood pixel. The threshold function $\theta(\cdot)$ is defined as:

$$\theta(t) = \begin{cases} 1 & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

LBP may lose intensity information due to the threshold function $\theta(\cdot)$. In order to tackle this issue, several approaches have been proposed, one of these approaches is NCDI which is proposed by Wu and Lin [4], NCDI is extracted on a grayscale image by iterating each pixel (x, y) and subtracting its value g_c from P neighboring pixel values $\{g_i\}_{i=1}^P$.

$$\text{NCDI}(x, y)_i = g_i - g_c \quad (3)$$

Finally, $\text{NCDI}_{i=1}^P$ were concatenated to create a multi-channel image. The P -channel of NCDI are extracted from a grayscale image will have edge information in P directions. Therefore, we propose to extract two types of NCDI histogram feature for texture classification.

- The first histogram feature is obtained from each NCDI by counting the frequency of each value from -255 to 255 . These histograms have information about the intensity of difference between pixel values.
- To further capture the edge information from NCDI, the second histogram feature is extracted from each NCDI by counting the frequency of 256 LBP values. These histograms have information about the correlation of difference between pixel values.

3 EXPERIMENTS

3.1 Dataset description

The proposed approaches are evaluated on four benchmark color texture datasets, including New BarkTex [5], Outex-TC-00013 [1], USPTex [6] and STex. Training and testing set of each dataset is divided by the holdout method (as shown in table 1).

Table 1. Summary of image datasets used in the experiment.

Dataset name	Image size	# class	# training	# test	Total
New BarkTex	64×64	6	816	816	1632
Outex-TC-00013	128×128	68	680	680	1360
USPTex	128×128	191	1146	1146	2292
STex	128×128	476	3808	3808	7616

3.2 Experimental setup

In order to evaluate the proposed approaches, experiments are conducted on the same training and testing set of four benchmark color texture datasets by using the nearest neighbor (1-NN) classifier associated with the L1 distance.

Firstly, the baseline result (LBP RGB) is obtained by extracting three $\text{LBP}_{8,1}$ histograms from the three channels of RGB images.

Secondly, two types of proposed histogram feature are evaluated separately. To begin, 8-channel NCDI is extracted from each channel of RGB image. Then, $\text{LBP}_{8,1}$ histograms of NCDI and histograms of NCDI are extracted from these NCDIs. Next, 1-NN classifier is used to obtain the accuracy of each type of feature.

Finally, two proposed types of histogram feature are concatenated produce the result of the proposed approach.

3.3 Results

Table 2. Classification accuracy (in %) of the original LBP approach and the proposed approaches on four texture datasets New BarkTex, Outex-TC-00013, USPTex, and STEX. LBP RGB stands for LBP histogram feature extracted on three channel of RGB image. LBP NCDI is the approach that uses the LBP histogram feature extracted on NCDIs. NCDI Histogram is the histogram feature extracted by counting the frequency of each value in NCDIs. NCDI Histogram & LBP NCDI is the proposed approach that concatenates two types of histogram features.

	New BarkTex	Outex-TC-00013	USPTex	STex
LBP RGB	76.6	86.0	85.3	85.5
LBP NCDI	74.4	88.9	82.3	87.8
NCDI Histogram	69.4	86.3	80.8	76.9
NCDI Histogram & LBP NCDI	78.3	90.2	88.7	89.1

Table 2 clearly shows that the combination of two proposed feature types outperforms the result of LBP histogram from RGB image. Comparing with features from LBP histograms of RGB image, the proposed approaches achieved significant gain of more than 1.7%, 4%, 3.4% and 3.5% in accuracy on New BarkTex, Outex-TC-00013, USPTex, and STEX dataset respectively.

Table 3 shows that our proposed approach to concatenate two types of feature obtains better results than several other approaches. According to the experiments on STEX dataset, the proposed approach outperforms other approaches by a margin of more than 1.5%. The classification results obtained on the USPTex dataset by the proposed method provides slightly better result than other approaches. In the case of Outex-TC-00013 dataset, the Mix color order LBP histogram approach give slightly better accuracy than ours. However, the proposed method outperforms that approach by improving 2.2%, 0.3%, and 2.1% on New BarkTex, USPTex and STEX datasets, respectively.

4 CONCLUSION

In this paper, two types of histogram features are proposed to extract edge information from NCDI. These two types of histograms are then concatenated to further capture useful information from NCDIs. In addition, edge features of NCDI have complementary information to LBP feature from RGB image. Therefore, histograms extracted from NCDI are concatenated with LBP histograms from RGB image to obtain the result of the proposed approach. Experimental results on four benchmark color texture datasets show that the concatenation of features from NCDI outperforms the original LBP with a large margin in

Table 3. Classification accuracy (in %) of the proposed method compared with other approaches on four texture datasets New BarkTex, Outex-TC-00013, USPTex, and STex.

Methods	New BarkTex	Outex-TC-00013	USPTex	STex
Color angles LBP [7]	71.0	86.2	79.1	-
Wavelet coefficients [8]	-	89.7	-	77.6
Color contrast occurrence matrix [9]	-	82.6	-	76.7
Soft color descriptors [10]	-	81.4	58.0	55.3
LBP and local color contrast [11]	71.0	85.3	82.9	-
CLBP [12]	72.8	84.4	72.3	-
Mix color order LBP histogram [13]	77.7	87.1	84.2	-
LTP [?]	76.1	90.6	88.4	87.0
LPQ [14]	66.2	81.4	86.6	87.6
TPLBP [?]	61.3	75.0	80.0	71.7
LBP Median [16]	72.3	83.0	84.1	81.9
NCDI Histogram & LBP NCDI	78.3	90.2	88.7	89.1

accuracy. Moreover, the proposed approach has achieved better results compare with several other approaches.

The extension of this work is to reduce the dimension of proposed features and apply it to other computer vision tasks.

References

1. T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, and S. Huovinen, “Outex - new framework for empirical evaluation of texture analysis algorithms,” in *Object recognition supported by user interaction for service robots*, vol. 1. Quebec City, Que., Canada: IEEE Comput. Soc, 2002, pp. 701–706.
2. L. Liu, P. Fieguth, Y. Guo, X. Wang, and M. Pietikäinen, “Local binary features for texture classification: Taxonomy and experimental study,” *Pattern Recognition*, vol. 62, pp. 135–160, Feb. 2017.
3. J. Lu, V. E. Liang, and J. Zhou, “Cost-sensitive local binary feature learning for facial age estimation,” vol. 24, no. 12, pp. 5356–5368, 2015.
4. B.-F. Wu and C.-H. Lin, “Adaptive Feature Mapping for Customizing Deep Learning Based Facial Expression Recognition Model,” *IEEE Access*, vol. 6, pp. 12 451–12 461, 2018.

5. A. Porebski, N. Vandenbroucke, L. Macaire, and D. Hamad, “A new benchmark image test suite for evaluating color texture classification schemes,” *Multimedia Tools and Applications*, vol. 70, 05 2014.
6. A. R. Backes, D. Casanova, and O. M. Bruno, “Color texture analysis based on fractal descriptors,” *Pattern Recognition*, vol. 45, no. 5, pp. 1984 – 1992, 2012.
7. A. Ledoux, O. Lossen, and L. Macaire, “Color local binary patterns: compact descriptors for texture classification,” *Journal of Electronic Imaging*, vol. 25, no. 6, p. 061404, 2016.
8. A. D. El Maliani, M. El Hassouni, Y. Berthoumieu, and D. Aboutajdine, “Color texture classification method based on a statistical multi-model and geodesic distance,” *J. Vis. Comun. Image Represent.*, vol. 25, no. 7, pp. 1717–1725, Oct. 2014.
9. A. Martinez Rios, N. Richard, and C. Fernandez-Maloigne, “Alternative to colour feature classification using colour contrast occurrence matrix,” vol. 9534, 06 2015.
10. R. Bello, F. Bianconi, A. Fernández, E. González, and F. Di Maria, “Experimental comparison of color spaces for material classification,” *Journal of Electronic Imaging*, vol. 25, p. 061406, 06 2016.
11. C. Cusano, P. Napoletano, and R. Schettini, “Combining local binary patterns and local color contrast for texture classification under varying illumination,” *Journal of the Optical Society of America A*, vol. 31, no. 7, p. 1453, Jul. 2014.
12. Z. Guo, L. Zhang, and D. Zhang, “A completed modeling of local binary pattern operator for texture classification,” *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657–1663, June 2010.
13. A. Ledoux, O. Lossen, and L. Macaire, “Color local binary patterns: Compact descriptors for texture classification,” *Journal of Electronic Imaging*, vol. 25, p. 061404, 05 2016.
14. V. Ojansivu, E. Rahtu, and J. Heikkila, “Rotation invariant local phase quantization for blur insensitive texture analysis,” in *2008 19th International Conference on Pattern Recognition*. Tampa, FL, USA: IEEE, Dec. 2008, pp. 1–4.
15. L. Wolf, T. Hassner, and Y. Taigman, “Descriptor Based Methods in the Wild,” p. 14.
16. A. Hafiane, K. Palaniappan, and G. Seetharaman, “Joint Adaptive Median Binary Patterns for texture classification,” *Pattern Recognition*, vol. 48, no. 8, pp. 2609–2620, Aug. 2015.

Feeding Convolutional Neural Network by hand-crafted features based on Enhanced Neighbor-Center Different Image for color texture classification

Duc Phan Van Hoai, Vinh Truong Hoang
Ho Chi Minh City Open University
97 Vo Van Tan Street, District 3. Ho Chi Minh City, Vietnam
e-mail: 1551010028duc@ou.edu.vn; vinh.th@ou.edu.vn

Abstract—Texture analysis has many important applications, including material recognition, face recognition, object detection, image segmentation. Local feature descriptors were the principle approach for texture analysis in the past. Recently, Convolutional Neural Network (CNN) has provided more promising results for texture recognition and other related computer vision tasks. Standard CNNs take labeled RGB images as input. However, other encoded images were used as an extra input to CNN, which have been shown that can improve the performance. We propose to feed the CNNs with the new encoded image. The experimental results on four benchmark color texture database show the efficiency of our proposed approach.

Keywords—*texture classification, deep learning, convolutional neural network, neighbor-center different image, color images, CNN*

I. INTRODUCTION

Texture analysis is an important field of computer vision with many real-life applications such as industrial and biomedical surface inspection, material and object classification, segmentation of satellite or aerial imagery. In reality, texture of the same material or object varies in illumination, orientation, scale, and rotation. Texture classification can be divided into two phases: features extraction and classification phase [1]. The first phase has attracted more attention because it needs a robust descriptor to represent and discriminate different classes. Many different approaches have been proposed to overcome these drawbacks in illumination, orientation, and other visual appearance problems. Most approaches are based on local descriptors and statistical analysis techniques.

Local Binary Pattern (LBP) introduced by Ojala et al. [2] is known as one of the most successful statistical approaches in many applications such as texture classification [3], [4], face recognition [5], age estimation [6]. It is a computational efficiency operator with high discriminative power and robustness against illumination. However, LBP may not work properly for noisy images due to its threshold function [7]. Many variants of LBP have been proposed to minimize LBP's limitation [8].

Convolutional Neural Network (CNN) has been introduced in 1989 [9], but it only got attention after winning the Ima-

geNet Large Scale Visual Recognition Challenge with a large margin [10]. Two main factors lead to this revolution are the appearance of enormous labeled datasets and massively parallel computing power of GPUs. After winning in 2012, Deep Learning has attracted many researchers around the world, a lot of techniques have been proposed to boost its performance. Therefore, Deep Learning approaches started surpassing traditional methods in many computer vision tasks [10]–[13]. Standard CNN models is a series of layers which trained on a large amount of labeled RGB images. There are three main types of layer in a CNN model: Convolutional Layer, Pooling Layer, and Fully-Connected (FC) Layer. Many efforts have been made to enhance the performance, including inventing new regularization techniques [14], [15], new activation functions [16], [17] and optimization algorithms [18], [19], designing new architectures [20]–[22] to let the gradient flow smoother and deeper into the network.

In recent years, many approaches are proposed to use multiple feature streams CNN models. They show the good performance in action recognition [23], [24] and texture recognition [25]. Different feature streams have complementary information so that fusing these feature streams into a single model will provide a better result. Several approaches apply a two-stream CNN model [23], [24] for action recognition task which composes the spatial and temporal stream. The first stream is trained on RGB images and the second stream is trained on dense optical flow images. These two streams are then trained separately and finally fuse into a single CNN model.

More recently, other approaches to feed CNN with hand-crafted features have successfully applied for computer vision tasks, including face recognition [26], facial expression recognition [27], [28], age estimation [29], and texture recognition [25], [30]. Nguyen et al. [26] investigated to fuse Multi-level Local Binary Pattern features and features extracted from CNN. Wu and Lin [28] proposed three types of Adaptive Feature Mapping to map features extracted from testing images to a space that closer to features space of training images, they also designed a new hand-crafted feature calls Neighbor-Center

Difference Image (NCDI), it is extracted by subtracting the center pixel from its neighboring pixels. Levi and Hassner [27] proposed to transform LBP images to LBP mapped coded images which are more suitable to feed for CNNs, they used an ensemble of CNN models trained on RGB images and LBP mapped coded images. Anwer et al. [25] extended Levi and Hassner's work by using a two-stream CNN model to fuse features extracted from LBP mapped coded images and RGB images. Hosseini et al. [29] fed CNN with Gabor response maps and gained excellent results in gender classification, age estimation, face detection, and emotion recognition.

Inspired by these successful approaches, we proposed to use an extended version of NCDI as the second input stream along with RGB input stream. Features extracted from two streams were fused by concatenating at the last Convolutional layer. Then, these features were fed into a series of FC layers and a Softmax output layer to classify into classes.

The rest of this paper is organized as follows. Section II introduces related works to our proposed approach. The pre-processing method and our proposed model are introduced in section III. We then present the experimental result on several benchmark color texture databases in section IV. Finally, the conclusion is discussed in section V.

II. RELATED WORK

Wu and Lin [28] feed pre-processing images as input for CNN models. The pre-processing process follows these steps. Extracting the landmarks on the face. Then, the ellipse cropping technique was used to remove unnecessary regions in images. Finally, 8-channel NCDI was extracted on the grayscale image. Wu and Lin manually designed a CNN model and trained it on pre-processed images. After CNN model was trained, features of all images in training set were extracted and stored as a feature database. In the testing phase, three types of Adaptive Feature Mapping (AFM) were used to map features of all images in the testing set so it is as close as features of the training set. In three types of AFM, Weighted Center Regression AFM always provides the best result. The use of pre-processing steps and AFM techniques have provided an excellent result on several facial expression datasets.

Hosseini et al. [29] used Gabor response maps combined with grayscale image as input for CNN models to deal with a wide range of face-related tasks. Eight different Gabor filters are applied to create eight Gabor response maps which will be concatenated with the grayscale input image to form a 9-channel input. Furthermore, they proposed to add a 1×1 Convolutional layer into the model to fuse 9-channel input. This approach has shown to provide better results on many face-related tasks, including gender classification, age estimation, face detection, and emotion recognition.

Levi and Hassner [27] proposed to transform LBP images to LBP mapped coded images which is more suitable to feed for CNNs. Firstly, code-to-code dissimilarity scores were computed on the binary array of all LBP values using an approximation to the Earth Mover's Distance, then Multidimensional Scaling was used to map LBP values to a 3D space using the dissimilarity matrix computed beforehand. LBP images extracted from grayscale images with different parameters were mapped to create LBP mapped coded images

which were used to feed for CNNs. They trained several CNN models on LBP mapped coded images with different parameter and many CNN models trained on RGB images. By using an assembly of 20 CNN models, the experimental results show a significant gain of 15.36% compare to the baseline result on Emotion Recognition in the Wild Challenge.

Anwer et al. [30] extended Levi and Hassner's work by fusing features extracted from CNN trained with LBP mapped coded images and CNN trained with RGB images. The model proposed by Anwer et al. was named TEX-Nets, they investigated early and late fusion architectures. Both fusion architectures have shown to provide better results compared to standard model train on RGB images. By the experimental process, the late fusion architecture always performs better than the early fusion architecture. They used two networks to create TEX-Nets late fusion model. Firstly, two networks were trained on ImageNet, one network was trained on RGB images from ImageNet dataset, the other one was trained on LBP mapped coded images of ImageNet, which was extracted the same way as Levi and Hassner. After pre-trained on ImageNet, two networks were fused by concatenating features of second last FC layer. TEX-Nets have outperformed State-of-the-art on several color texture datasets.

III. METHOD

Motivated by the success of Wu and Lin by using edge information in facial expression recognition [28] and the excellent results in texture classification provided by multiple feature streams CNN models [30], we investigated to use a two-stream CNN model. The purpose is to fuse features extracted from RGB images and enhanced NCDI feature. The pre-processing method and the model architecture are described in the following subsections.

A. Pre-processing methodology

LBP is a powerful local descriptor to deal with texture classification and face related tasks. The definition of the original LBP operator has then been generalized to explore intensity values of points on a circular neighborhood. The circular neighborhood is defined by considering the values of radius R and P neighbors around the central pixel. The $\text{LBP}_{P,R}$ code is computed by comparing the gray value g_c of the central pixel with the gray values $\{g_i\}_{i=0}^{P-1}$ of its P neighbors, as follows:

$$\text{LBP}_{P,R} = \sum_{i=0}^{P-1} s(g_i - g_c) \times 2^i \quad (1)$$

where the threshold function $s(t)$ is defined as:

$$s(t) = \begin{cases} 1 & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

LBP may lose magnitude information due to the threshold function $s(t)$. In order to tackle this issue, Wu and Lin proposed NCDI [28], which is extracted on a grayscale image by iterating each pixel (x, y) and subtracting its value g_c from P neighboring pixel values $\{g_i\}_{i=1}^P$.

$$\text{NCDI}(x, y)_i = g_i - g_c \quad (3)$$

Finally, $\text{NCDI}_{i=1}^P$ were concatenated to create a multi-channel image. The 8-channel NCDI is extracted from a grayscale image will have information about edges in 8 directions. We propose to use NCDI for texture classification. All texture images firstly are transformed to grayscale. Then, 8-channel NCDI are extracted. In fact, NCDI mainly stores edge information, the grayscale image was combined with 8-channel NCDI to create an Enhanced NCDI (ENCDI), which contains information of edges and illumination (as shown in figure 1).

B. Proposed Model

Our two-stream model was created with two VGG-16 networks [31], one network was pre-trained on ImageNet using RGB images. A Global Average Pooling layer was added after the last Max-pooling layer of each VGG-16 model to reduce overfitting. Then, three FC layers and a Softmax layer were added. In front of each new FC layers, a Dropout layer with a dropout rate of 50% was inserted. Figure 2 illustrates the proposed two-stream model composed of two VGG-16.

The following section presents the experimental results of the proposed approach.

IV. EXPERIMENTS

A. Dataset description

The proposed method is evaluated on four benchmark color texture datasets such as New BarkTex [32], Outex-TC-00013 [2], USPTex [33] and STex. Each dataset is divided into training set and testing set by holdout method (as shown in table I).

TABLE I: Summary of image databases used in experiment.

Dataset name	Image size	# class	# training	# test	Total
New BarkTex	64×64	6	816	816	1632
Outex-TC-00013	128×128	68	680	680	1360
USPTex	128×128	191	1146	1146	2292
STex	128×128	476	3808	3808	7616

B. Experimental setup

The models were implemented using Keras framework. Training and testing were performed on Google Cloud Computing Platform with a configuration of 8 CPUs 2.5 GHz, 52 GBs of RAM, 1 NVIDIA Tesla V100 GPU 16 GBs RAM.

Multi-crop technique was used as a form of data augmentation. Five regions from each image were cropped as follow: four regions aligned with the four corners of the input image and one from the center. At the training phase, these five regions from each image of the training set were used as input image for the model. At test time, the predictions of five regions from the same original image were averaged to produce the final prediction. Multi-crop technique was not used on the New BarkTex dataset due to its low resolution.

Five regions with a size of 96×96 pixels were cropped on Outex-TC-00013, USPTex, and STex dataset.

For fair comparison, all models were fine-tuned with the same hyperparameters. The batch size for New BarkTex is 16 and 64 for other datasets. All models used Adam optimizer [18] with a learning rate of 10^{-5} , and then decreased by a factor of 10 when the validation set accuracy stopped improving on the first time. On the second time, the learning rate is remained 10^{-6} and the batch size is set to 256.

In order to provide the VGG-16 RGB result, the VGG-16 pre-trained on RGB images of ImageNet is used. Three FC layers of the pre-trained model are replaced with new FC layers. A dropout layer with drop ratio of 0.5 is added after each of the first two FC layers as described in [31]. The model was trained with the same hyperparameters as other models.

The two-stream model was implemented as described in subsection B of section III. the model was trained with the same hyperparameters as the other models to produce the results on of four benchmark color texture datasets.

The baseline model which is trained on cropped RGB images with size 96×96 pixels has 50,653,060 parameters. The proposed two-stream model trained on cropped RGB images and cropped ENCDI has 50,691,140 parameters. On OuTex-TC-00013 dataset, to train on 3400 cropped images with size 96×96 pixels, the baseline model and the proposed model take approximately 6.3 and 12.1 seconds respectively.

C. Results

Table II shows the result of our proposition and compares with other methods on four considered datasets. VGG-16 RGB represents the VGG-16 model trains only on RGB images, and the VGG-16 RGB & ENCDI represents the two-stream model trains on RGB and ENCDI. It is worth to note that, we only collect and compare our result with the results using holdout method for dividing training and testing set in the state-of-the-art.

From table II, we can observe that the proposed approach consistently outperform the VGG-16 RGB approach. It shows that ENCDI has useful information. Globally, VGG-16 RGB & ENCDI approach provides better results than other methods on New BarkTex, USPTex and STex dataset.

On the New BarkTex dataset, due to its low resolution, multi-crop technique is not used. However, the deep learning models have outperformed other approaches by a margin of more than 1.8%. VGG-16 RGB & ENCDI achieves a mean classification accuracy of 95.6% which is 1.2% better than VGG-16 RGB.

In the case of Outex-TC-00013 dataset, the local descriptor approaches give better performance than ours. However, the proposed model outperforms the latest results in the literature by improving more than 3%, 2%, 0.9% on New BarkTex, USPTex and STex datasets, respectively.

According to the experiments on USPTex dataset, our models give better results by improving the accuracy more than 1.7%. Similarly, the classification results obtain on the STex dataset by the Deep Learning models are 97.6%, and

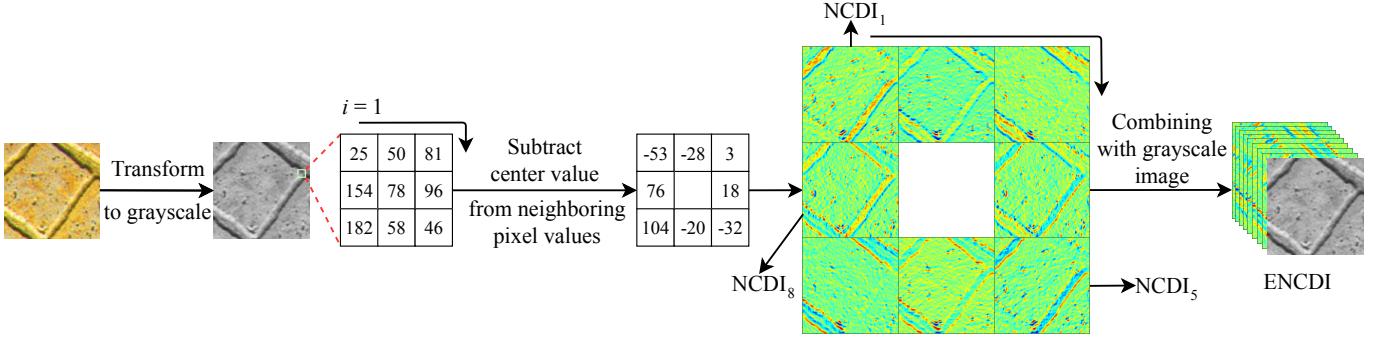


Fig. 1: An illustration of pre-processing method to obtain ENCDI.

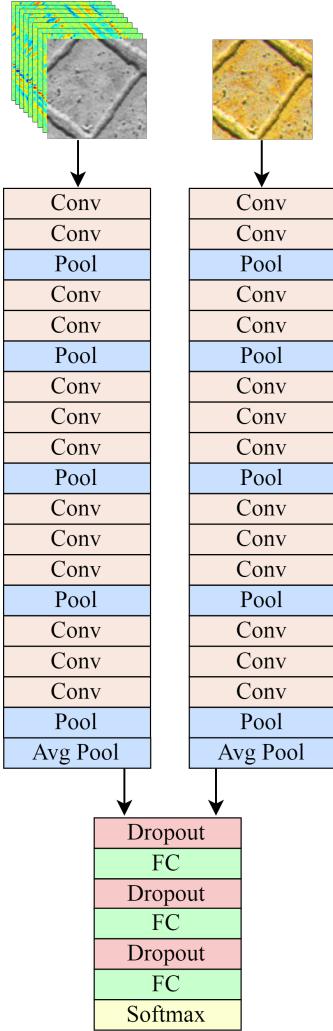


Fig. 2: The proposed two-stream model composed of multiple VGG-16. FC is Fully-Connected layer, Conv stands for Convolutional layer, Pool is Max-pooling layer, and Avg Pool is Global Average Pooling layer.

96.2% corresponds to VGG-16 RGB & ENCDI, and VGG-16 RGB respectively. The proposed method provides slightly

TABLE II: Classification accuracy (in %) of the proposed method compared with other approaches on four texture datasets New BarkTex, Outex-TC-00013, USPTex, and STex in the state-of-the-art.

Methods	New BarkTex	Outex-TC-00013	USPTex	STex
Wavelet coefficients [34]	-	89.7	-	77.6
Color contrast occurrence matrix [35]	-	82.6	-	76.7
Combine color and LBP-based features [36]	-	90.2	95.7	-
Quaternion-Michelson Descriptor [37]	-	91.3	94.2	-
Halftoning Local Derivative Pattern and Color Histogram [38]	-	88.2	93.9	-
3D Color histogram [39]	79.9	94.7	-	-
Soft color descriptors [40]	-	81.4	58.0	55.3
LBP and local color contrast [41]	71.0	85.3	82.9	-
CLBP [42]	72.8	84.4	72.3	-
Mix color order LBP histogram [43]	77.7	87.1	84.2	-
Color angles [44]	80.2	86.2	88.8	-
Sparse score [45]	81.3	93.4	93.2	-
DRLBP [46]	61.4	89.0	89.4	89.4
MCSBS [47]	87.8	92.9	97.3	96.7
ASL-based MCSHS [47]	86.8	95.3	97.6	96.1
ICS-based MCSHS [47]	92.6	95.6	97.2	94.1
VGG-16 RGB	94.4	93.5	99.3	96.2
VGG-16 RGB & ENCDI (Our proposed method)	95.6	94.7	99.6	97.6

better results than other approaches.

V. CONCLUSION

In this paper, we propose to feed ENCDI as the second input stream along with the RGB input stream. The ENCDI gets edge features from NCDI and illumination information from grayscale image. Experimental results have shown that our proposed approach can improve the accuracy of the color texture classification task. The future of this work is now extended to combine other features to fully capture local and global information.

REFERENCES

- [1] M. Mirmehdi, X. Xie, and J. Suri, *Handbook of Texture Analysis*. London, UK, UK: Imperial College Press, 2009.
- [2] T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, and S. Huovinen, “Outex - new framework for empirical evaluation of texture analysis algorithms,” in *Object recognition supported by user interaction for service robots*, vol. 1. Quebec City, Que., Canada: IEEE Comput. Soc, 2002, pp. 701–706.
- [3] L. Liu, L. Zhao, Y. Long, G. Kuang, and P. Fieguth, “Extended local binary patterns for texture classification,” *Image and Vision Computing*, vol. 30, no. 2, pp. 86–99, Feb. 2012.
- [4] T. Mäenpää and M. Pietikäinen, “Texture analysis with local binary patterns,” in *Handbook of Pattern Recognition and Computer Vision*, 3rd ed. World Scientific, Jan. 2005, pp. 197–216.
- [5] T. Ahonen and A. Hadid, “Face Recognition with Local Binary Patterns,” in *Computer Vision - ECCV 2004*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, vol. 3021, pp. 469–481.
- [6] J. Ylioinas, A. Hadid, X. Hong, and M. Pietikäinen, “Age Estimation Using Local Binary Pattern Kernel Density Estimate,” in *Image Analysis and Processing – ICIAP 2013*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, vol. 8156, pp. 141–150.
- [7] X.-H. Han, G. Xu, and Y.-W. Chen, “Robust local ternary patterns for texture categorization,” in *2013 6th International Conference on Biomedical Engineering and Informatics*, pp. 846–850, 2013.
- [8] L. Liu, P. Fieguth, Y. Guo, X. Wang, and M. Pietikäinen, “Local binary features for texture classification: Taxonomy and experimental study,” *Pattern Recognition*, vol. 62, pp. 135–160, Feb. 2017.
- [9] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation Applied to Handwritten Zip Code Recognition,” *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec. 1989.
- [10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
- [11] M. Cimpoi, S. Maji, and A. Vedaldi, “Deep filter banks for texture recognition and segmentation,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, Jun. 2015, pp. 3828–3836.
- [12] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, Jun. 2015, pp. 815–823.
- [13] E. Shelhamer, J. Long, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, Apr. 2017.
- [14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.
- [15] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML’15. JMLR.org, 2015, pp. 448–456.
- [16] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML’10. USA: Omnipress, 2010, pp. 807–814.
- [17] A. L. Maas, “Rectifier nonlinearities improve neural network acoustic models,” 2013.
- [18] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations*, 12 2014.
- [19] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *Proceedings of COMPSTAT’2010*, Y. Lechevallier and G. Saporta, Eds. Heidelberg: Physica-Verlag HD, 2010, pp. 177–186.
- [20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” pp. 1–9, June 2015.
- [21] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” pp. 770–778, June 2016.
- [22] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017.
- [23] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” pp. 568–576, 2014.
- [24] G. Cheron, I. Laptev, and C. Schmid, “P-CNN: Pose-Based CNN Features for Action Recognition,” in *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile: IEEE, Dec. 2015, pp. 3218–3226.
- [25] R. M. Anwer, F. S. Khan, J. van de Weijer, M. Molinier, and J. Laaksonen, “Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 138, pp. 74–85, Apr. 2018.
- [26] D. T. Nguyen, T. D. Pham, N. R. Baek, and K. R. Park, “Combining Deep and Handcrafted Image Features for Presentation Attack Detection in Face Recognition Systems Using Visible-Light Camera Sensors,” *Sensors*, vol. 18, no. 3, p. 699, Feb. 2018.
- [27] G. Levi and T. Hassner, “Emotion Recognition in the Wild via Convolutional Neural Networks and Mapped Binary Patterns.” ACM Press, 2015, pp. 503–510.
- [28] B.-F. Wu and C.-H. Lin, “Adaptive Feature Mapping for Customizing Deep Learning Based Facial Expression Recognition Model,” *IEEE Access*, vol. 6, pp. 12 451–12 461, 2018.
- [29] S. Hosseini, S. H. Lee, and N. I. Cho, “Feeding hand-crafted features for enhancing the performance of convolutional neural networks,” *CoRR*, vol. abs/1801.07848, 2018.
- [30] F. S. Khan, J. van de Weijer, and J. Laaksonen, “TEX-Nets: Binary Patterns Encoded Convolutional Neural Networks for Texture Recognition.” ACM Press, 2017, pp. 125–132.
- [31] S. Liu and W. Deng, “Very deep convolutional neural network based image classification using small training sample size,” in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*. Kuala Lumpur, Malaysia: IEEE, Nov. 2015, pp. 730–734.
- [32] A. Porebski, N. Vandebroucke, L. Macaire, and D. Hamad, “A new benchmark image test suite for evaluating color texture classification schemes,” *Multimedia Tools and Applications*, vol. 70, 05 2014.
- [33] A. R. Backes, D. Casanova, and O. M. Bruno, “Color texture analysis based on fractal descriptors,” *Pattern Recognition*, vol. 45, no. 5, pp. 1984 – 1992, 2012.
- [34] A. D. El Maliani, M. El Hassouni, Y. Berthoumieu, and D. Aboutajdine, “Color texture classification method based on a statistical multi-model and geodesic distance,” *J. Vis. Comun. Image Represent.*, vol. 25, no. 7, pp. 1717–1725, Oct. 2014.
- [35] A. Martinez Rios, N. Richard, and C. Fernandez-Maloigne, “Alternative to colour feature classification using colour contrast occurrence matrix,” vol. 9534, 06 2015.
- [36] P. Liu, J.-M. Guo, K. Chamnongthai, and H. Prasetyo, “Fusion of color histogram and lbp-based features for texture image retrieval and classification,” *Information Sciences*, vol. 390, pp. 95 – 111, 2017.
- [37] R. Lan and Y. Zhou, “Quaternion-michelson descriptor for color image classification,” *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5281–5292, Nov 2016.
- [38] J.-M. Guo, H. Prasetyo, H. Lee, and C.-C. Yao, “Image retrieval using indexed histograms of void-and-cluster block truncation coding,” *Signal Processing*, vol. 123, pp. 143 – 156, 2016.
- [39] M. Pietikäinen, T. Mäenpää, and J. Viertola, “Color texture classification with color histograms and local binary patterns,” *Workshop on Texture Analysis in Machine Vision*, 01 2002.
- [40] R. Bello, F. Bianconi, A. Fernández, E. González, and F. Di María, “Experimental comparison of color spaces for material classification,” *Journal of Electronic Imaging*, vol. 25, p. 061406, 06 2016.
- [41] C. Cusano, P. Napoletano, and R. Schettini, “Combining local binary patterns and local color contrast for texture classification under varying illumination,” *Journal of the Optical Society of America A*, vol. 31, no. 7, p. 1453, Jul. 2014.
- [42] Z. Guo, L. Zhang, and D. Zhang, “A completed modeling of local binary

- pattern operator for texture classification," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657–1663, June 2010.
- [43] A. Ledoux, O. Lossen, and L. Macaire, "Color local binary patterns: Compact descriptors for texture classification," *Journal of Electronic Imaging*, vol. 25, p. 061404, 05 2016.
 - [44] S. H. Lee, J. Y. Choi, Y. M. Ro, and K. N. Plataniotis, "Local color vector binary patterns from multichannel face images for face recognition," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2347–2353, April 2012.
 - [45] V. T. Hoang, A. Porebski, N. Vandenbroucke, and D. Hamad, "LBP histogram selection based on sparse representation for color texture classification;" in *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science and Technology Publications, pp. 476–483.
 - [46] R. Mehta and K. Egiazarian, "Dominant rotated local binary patterns (drlbp) for texture classification," *Pattern Recognition Letters*, vol. 71, pp. 16 – 22, 2016.
 - [47] A. Porebski, V. T. Hoang, N. Vandenbroucke, and D. Hamad, "Multi-color space local binary pattern-based feature selection for texture classification," *Journal of Electronic Imaging*, vol. 11010, no. 1, 2018.

Tài liệu tham khảo

- [1] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” in *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*. ACM Press, pp. 1–8.
- [2] T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, and S. Huovinen, “Outex - new framework for empirical evaluation of texture analysis algorithms,” in *Object recognition supported by user interaction for service robots*, vol. 1. Quebec City, Que., Canada: IEEE Comput. Soc, 2002, pp. 701–706.
- [3] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” vol. 24, no. 7, pp. 971–987.
- [4] Xiaoyang Tan and B. Triggs, “Enhanced local texture feature sets for face recognition under difficult lighting conditions,” vol. 19, no. 6, pp. 1635–1650.
- [5] R. Mehta and K. Egiazarian, “Rotated local binary pattern (rlbp): Rotation invariant texture descriptor,” in *2nd International Conference on Pattern Recognition Applications and Methods, ICPRAM 2013, Barcelona, Spain, 15.-18.2.2013*, ser. International Conference on Pattern Recognition Applications and Methods. Institute of Electrical and Electronics Engineers IEEE, 2013, pp. 497–502, institute of Electrical and Electronics Engineers IEEE.
- [6] Z. Guo, L. Zhang, and D. Zhang, “A completed modeling of local binary pattern operator for texture classification,” *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657–1663, June 2010.
- [7] C. Cortes and V. Vapnik, “Support-vector networks,” vol. 20, no. 3, pp. 273–297.
- [8] B.-F. Wu and C.-H. Lin, “Adaptive feature mapping for customizing deep learning based facial expression recognition model,” vol. 6, pp. 12 451–12 461.
- [9] S. Liu and W. Deng, “Very deep convolutional neural network based image classification using small training sample size,” in *2015 3rd IAPR Asian Conference*

on Pattern Recognition (ACPR). Kuala Lumpur, Malaysia: IEEE, Nov. 2015, pp. 730–734.

- [10] A. Porebski, N. Vandenbroucke, L. Macaire, and D. Hamad, “A new benchmark image test suite for evaluating color texture classification schemes,” *Multimedia Tools and Applications*, vol. 70, 05 2014.
- [11] A. R. Backes, D. Casanova, and O. M. Bruno, “Color texture analysis based on fractal descriptors,” *Pattern Recognition*, vol. 45, no. 5, pp. 1984 – 1992, 2012.
- [12] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations*, 12 2014.
- [13] A. Ledoux, O. Lossen, and L. Macaire, “Color local binary patterns: compact descriptors for texture classification,” *Journal of Electronic Imaging*, vol. 25, no. 6, p. 061404, 2016.
- [14] A. D. El Maliani, M. El Hassouni, Y. Berthoumieu, and D. Aboutajdine, “Color texture classification method based on a statistical multi-model and geodesic distance,” *J. Vis. Comun. Image Represent.*, vol. 25, no. 7, pp. 1717–1725, Oct. 2014.
- [15] A. Martinez Rios, N. Richard, and C. Fernandez-Maloigne, “Alternative to colour feature classification using colour contrast occurrence matrix,” vol. 9534, 06 2015.
- [16] R. Bello, F. Bianconi, A. Fernández, E. González, and F. Di Maria, “Experimental comparison of color spaces for material classification,” *Journal of Electronic Imaging*, vol. 25, p. 061406, 06 2016.
- [17] C. Cusano, P. Napoletano, and R. Schettini, “Combining local binary patterns and local color contrast for texture classification under varying illumination,” *Journal of the Optical Society of America A*, vol. 31, no. 7, p. 1453, Jul. 2014.
- [18] A. Ledoux, O. Lossen, and L. Macaire, “Color local binary patterns: Compact descriptors for texture classification,” *Journal of Electronic Imaging*, vol. 25, p. 061404, 05 2016.
- [19] V. Ojansivu, E. Rahtu, and J. Heikkila, “Rotation invariant local phase quantization for blur insensitive texture analysis,” in *2008 19th International Conference on Pattern Recognition*. Tampa, FL, USA: IEEE, Dec. 2008, pp. 1–4.
- [20] L. Wolf, T. Hassner, and Y. Taigman, “Descriptor Based Methods in the Wild,” p. 14.

- [21] A. Hafiane, K. Palaniappan, and G. Seetharaman, “Joint Adaptive Median Binary Patterns for texture classification,” *Pattern Recognition*, vol. 48, no. 8, pp. 2609–2620, Aug. 2015.
- [22] P. Liu, J.-M. Guo, K. Chamnongthai, and H. Prasetyo, “Fusion of color histogram and lbp-based features for texture image retrieval and classification,” *Information Sciences*, vol. 390, pp. 95 – 111, 2017.
- [23] R. Lan and Y. Zhou, “Quaternion-michelson descriptor for color image classification,” *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5281–5292, Nov 2016.
- [24] J.-M. Guo, H. Prasetyo, H. Lee, and C.-C. Yao, “Image retrieval using indexed histogram of void-and-cluster block truncation coding,” *Signal Processing*, vol. 123, pp. 143 – 156, 2016.
- [25] M. Pietikäinen, T. Mäenpää, and J. Viertola, “Color texture classification with color histograms and local binary patterns,” *Workshop on Texture Analysis in Machine Vision*, 01 2002.
- [26] S. H. Lee, J. Y. Choi, Y. M. Ro, and K. N. Plataniotis, “Local color vector binary patterns from multichannel face images for face recognition,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2347–2353, April 2012.
- [27] V. T. Hoang, A. Porebski, N. Vandenbroucke, and D. Hamad, “LBP histogram selection based on sparse representation for color texture classification;,” in *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science and Technology Publications, pp. 476–483.
- [28] R. Mehta and K. Egiazarian, “Dominant rotated local binary patterns (drlbp) for texture classification,” *Pattern Recognition Letters*, vol. 71, pp. 16 – 22, 2016.
- [29] A. Porebski, V. T. Hoang, N. Vandenbroucke, and D. Hamad, “Multi-color space local binary pattern-based feature selection for texture classification,” *Journal of Electronic Imaging*, vol. 11010, no. 1, 2018.