

**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN CUỐI KÌ**

**ĐỀ TÀI**  
**NHẬN DẠNG VÀ PHÂN LOẠI HÌNH ẢNH CÁC LOẠI**  
**CÔN TRÙNG GÂY HẠI NÔNG NGHIỆP**

Giảng viên hướng dẫn: Mai Tiến Dũng

Sinh viên thực hiện:

Phan Thanh Hải	22520390
Trương Hồng Anh	22520084
Nguyễn Hải Đăng	22520189

**TP.Hồ Chí Minh, 29 tháng 05 năm 2024**

## MỤC LỤC

<b>PHẦN 1: GIỚI THIỆU .....</b>	<b>4</b>
1. Lý do chọn đề tài.....	4
2. Xác định bài toán.....	4
2.1. Input .....	4
2.2. Output .....	4
<b>PHẦN 2: PHƯƠNG PHÁP .....</b>	<b>4</b>
1. Trích xuất đặc trưng .....	4
1.1. Histogram.....	4
1.1.1. Định nghĩa .....	4
1.1.2. Hướng tiếp cận .....	5
1.2. Resnet18 .....	6
1.2.1. Định nghĩa .....	6
1.2.2. Hướng tiếp cận .....	6
2. Huấn luyện mô hình .....	7
2.1. SVM.....	7
2.2. KNN.....	8
2.3. Softmax Regression.....	9
<b>PHẦN 3: THỰC NGHIỆM.....</b>	<b>10</b>
1. Bộ dữ liệu.....	10
2. Độ đo .....	10
3. Kết quả.....	11
3.1. So sánh tham số bins của phương pháp HSV Histogram .....	11
3.2. So sánh siêu tham số mô hình SVM bằng phương pháp Grid Search.....	12
3.3. So sánh siêu tham số mô hình KNN bằng phương pháp Grid Search.....	14

3.4. So sánh siêu tham số mô hình Softmax Regression bằng phương pháp Grid Search .....	15
3.5. So sánh kết quả.....	17
<b>Phần 4: Đánh giá.....</b>	<b>18</b>
1. Ưu và nhược điểm của đặc trưng Hsv histogram.....	18
1.1. Ưu điểm .....	18
1.2. Nhược điểm.....	18
2. ưu và nhược điểm của đặc trưng resnet.....	18
2.1. Ưu điểm.....	18
2.2. Nhược điểm.....	19
<b>PHẦN 4: TỔNG KẾT .....</b>	<b>20</b>
1. bài toán .....	20
2. mức độ hoàn thành .....	20
<b>PHẦN 5: TÀI LIỆU THAM KHẢO .....</b>	<b>21</b>

## PHẦN 1: GIỚI THIỆU

### 1. LÝ DO CHỌN ĐỀ TÀI

Đề tài phân loại các loài côn trùng gây hại cho nông nghiệp đóng góp lớn vào việc bảo vệ mùa màng và tăng suất nông nghiệp. Việc phát hiện chính xác côn trùng gây hại cho phép nông dân áp dụng các biện pháp phòng ngừa kịp, giảm thiểu thiệt hại kinh tế và tối ưu hóa việc sử dụng tài nguyên. Hiện nay, vô số các ứng dụng thực tế trên khắp thế giới đã và đang áp dụng thành công giải pháp từ bài toán này. Bên cạnh đó, các ngành học liên quan đến nông nghiệp cũng chiếm một tỉ trọng không nhỏ trong giáo dục bậc đại học và cao đẳng. Điều đó là minh chứng cho tầm quan trọng cho việc nghiên cứu đề tài này.

### 2. XÁC ĐỊNH BÀI TOÁN

#### 2.1. Input

Ảnh số miêu tả một trong 8 loại côn trùng gây hại cho nông nghiệp bao gồm: bọ rùa 28 chấm, sâu xám, bọ xít nâu dài, sâu bắp cải, bọ xít hút máu, ruồi Địa Trung Hải, bọ xít, bọ nhảy. Không gian màu ảnh số là RGB.

#### 2.2. Output

Nhãn phân loại thuộc 8 loại côn trùng gây hại cho nông nghiệp trên.

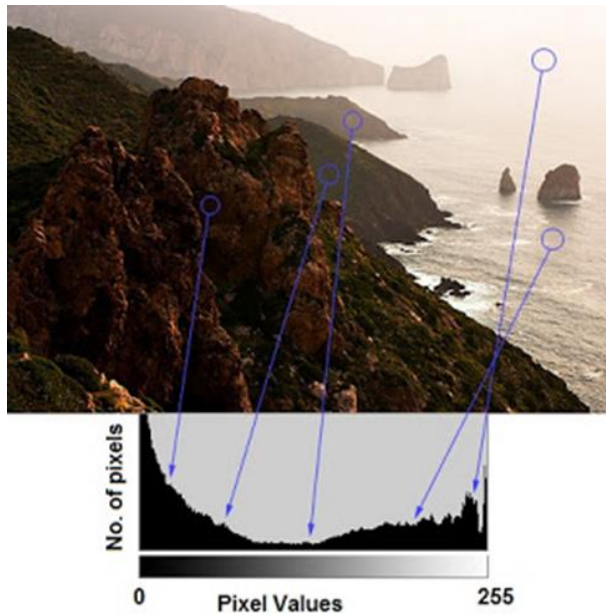
## PHẦN 2: PHƯƠNG PHÁP

### 1. TRÍCH XUẤT ĐẶC TRƯNG

#### 1.1. Histogram

##### 1.1.1. Định nghĩa

*Histogram* là một biểu đồ thống kê mô tả phân bố của dữ liệu. Nó thể hiện tần suất xuất hiện của các giá trị (từ 0 đến 255) trong một tập dữ liệu.



Trong biểu đồ trên, số lượng điểm ảnh tối nhiều hơn số lượng điểm ảnh sáng và lượng âm trung (giá trị pixel ở tầm trung, khoảng 127) là rất ít.

### 1.1.2. Hướng tiếp cận

Sử dụng ảnh ở không gian màu *HSV* thay cho *RGB* vì trong không gian màu *HSV*, kênh *Hue* (màu sắc) độc lập với độ sáng và độ bão hòa. Điều này giúp phân biệt màu sắc dễ dàng hơn so với không gian màu *RGB*, nơi thông tin màu và độ sáng được trộn lẫn.



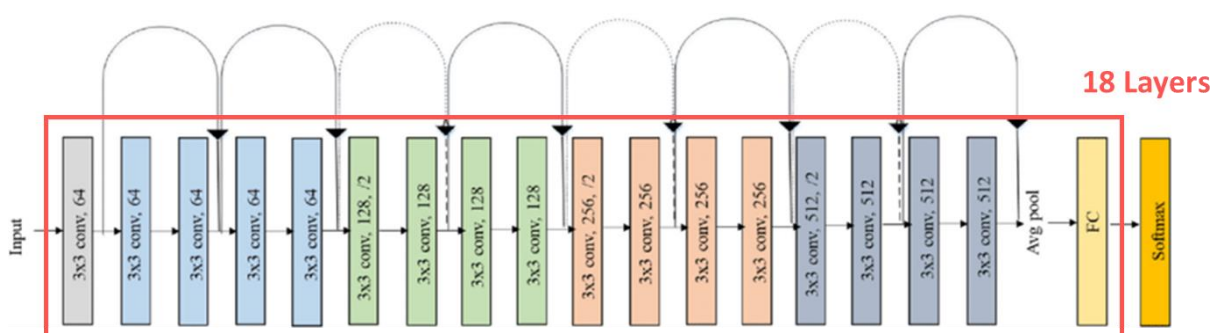
Thay vì tính tần suất xuất hiện ở từng điểm ảnh thì chia thành từng khoảng nhất định và giá trị của nó là tần suất xuất hiện của các điểm ảnh nằm trong khoảng đó. Mỗi khoảng giá trị như vậy được gọi là *bin*.

## 1.2. Resnet18

### 1.2.1. Định nghĩa

*ResNet (Residual Network)* là một kiến trúc mạng nơ-ron tích chập được giới thiệu bởi *Kaiming He* và đồng nghiệp vào năm 2015 và giành được vị trí thứ 1 trong cuộc thi *ILSVRC 2015* với tỉ lệ lỗi top 5 chỉ **3.57%**. Nó còn đứng vị trí đầu tiên trong cuộc thi *ILSVRC and COCO 2015* với *ImageNet Detection*, *ImageNet localization*, *Coco detection* và *Coco segmentation*.

*ResNet18* có **18 lớp (layer)** và sử dụng kỹ thuật *skip connection* để giảm hiện tượng *vanishing gradient* trong quá trình huấn luyện. Kiến trúc này đã đạt được hiệu suất tốt trong nhiều nhiệm vụ nhận dạng ảnh.



### 1.2.2. Hướng tiếp cận

Chuẩn hóa hình ảnh đầu vào theo như tập dữ liệu huấn luyện của *ResNet* để đảm bảo tính tương thích giữa dữ liệu đầu vào và trọng số đã được huấn luyện của mô hình, giúp cải thiện hiệu suất và tăng tốc quá trình học chuyển giao.

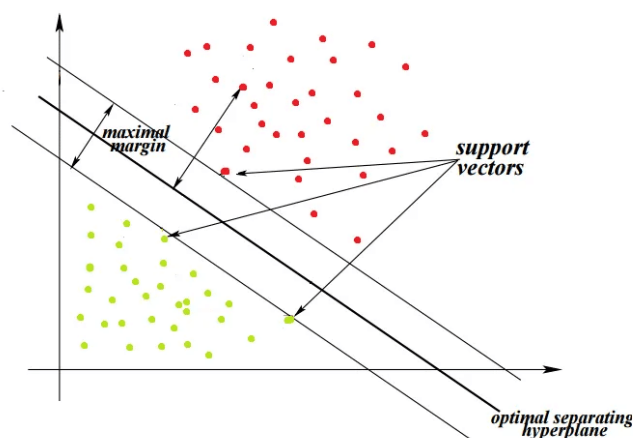


Phương pháp xuất trích trên sử dụng thông số mặc định của *pretrained ResNet18* và loại bỏ lớp *fully-connected(FC)* cuối cùng. Đầu ra là một vector có **512** chiều và lấy nó làm vector đặc trưng.

## 2. HUẤN LUYỆN MÔ HÌNH

### 2.1. SVM

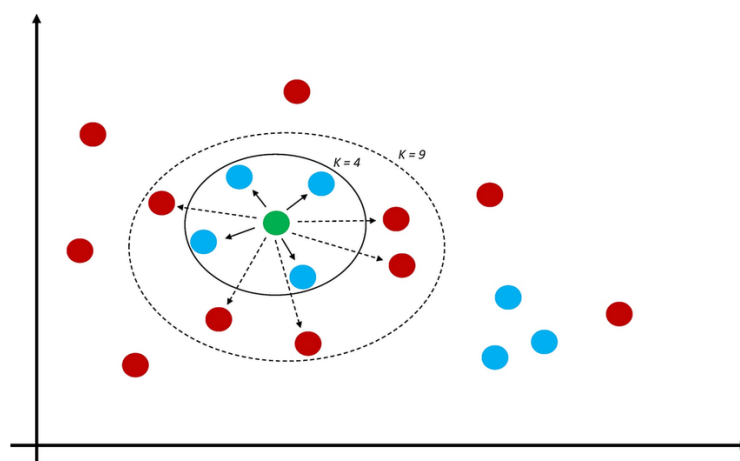
*Support Vector Machine (SVM)* được sáng tạo bởi *Vladimir N. Vapnik* và *Alexey Ya. Chervonenkis* vào năm 1964. Sau đó, vào năm 1992, *Bernhard Boser*, *Isabelle Guyon* và *Vladimir Vapnik* đề xuất phương pháp tạo mặt phẳng phân cách dữ liệu không tuyến tính bằng cách áp dụng *thủ thuật Kernel (Kernel Trick)*.



Mục tiêu của *SVM* là tìm ra một siêu phẳng trong không gian **N chiều** (ứng với **N** đặc trưng) chia dữ liệu thành hai phần tương ứng với lớp của chúng. Nói theo ngôn ngữ của đại số tuyến tính, siêu phẳng này phải có lẽ cực đại và phân chia hai bao lồi và cách đều chúng. Trong không gian **N chiều**, một siêu phẳng là một không gian con có kích thước **N-1 chiều**. Một cách trực quan, trong một mặt phẳng (2 chiều) thì siêu phẳng là một đường thẳng, trong một không gian 3 chiều thì siêu phẳng là một mặt phẳng. Mục tiêu của chúng ta là tìm ra siêu phẳng có lẽ rộng nhất tức là có khoảng cách tới các điểm của hai lớp là lớn nhất.

## 2.2. KNN

*K-Nearest Neighbors (KNN)* là thuật toán học giám sát được giới thiệu lần đầu bởi *Evelyn Fix* và *Joseph Hodges* vào năm 1951, sau đó được phát triển bởi *Thomas Cover*.



*KNN* hoạt động dựa trên nguyên tắc rằng các điểm dữ liệu tương tự nhau thường nằm gần nhau. Để dự đoán nhãn cho một điểm dữ liệu mới, *KNN* xem xét ‘**K**’ điểm dữ liệu gần nhất và dựa trên đa số nhãn của chúng để quyết định nhãn cho điểm dữ liệu mới đó. Giá trị ‘**K**’ là một siêu tham số quan trọng – nó quy định số lượng điểm dữ liệu gần nhất mà thuật toán sẽ xem xét. Lựa chọn **K** phù hợp là cực kỳ quan trọng: một giá trị **K** quá nhỏ có thể dẫn đến mô hình bị ảnh hưởng bởi nhiễu dữ liệu, trong khi một giá trị **K** quá lớn có thể làm mô hình không nhạy với các đặc điểm cụ thể của dữ liệu. Việc tính toán khoảng cách giữa các điểm dữ liệu là một phần quan trọng của *KNN*. Có nhiều phương



pháp để tính khoảng cách, nhưng phổ biến nhất là khoảng cách *Euclidean*, mặc dù các phương pháp khác như khoảng cách *Manhattan* cũng được sử dụng tùy thuộc vào bản chất của dữ liệu.

### 2.3. Softmax Regression

*Softmax Regression*, còn được biết đến với tên gọi là *Multinomial Logistic Regression*, là một phương pháp mở rộng của *Logistic Regression* và đã được phát triển để giải quyết các bài toán phân loại nhiều lớp. Hàm *Softmax* được biết đến với tên gọi *phân phối Boltzmann* và *phân phối Gibbs* được trình bày bởi nhà vật lý học và triết học người Úc *Ludwig Boltzmann* năm 1868.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

Hàm *Softmax* tính toán xác suất xảy ra của một sự kiện. Nói một cách khái quát, hàm *Softmax* sẽ tính khả năng xuất hiện của một phân lớp trong tổng số tất cả các phân lớp có thể xuất hiện. Sau đó, xác suất này sẽ được sử dụng để xác định phân lớp mục tiêu cho các dữ liệu đầu vào.

Cụ thể, hàm *Softmax* biến vector **K chiều** có các giá trị thực bất kỳ thành vector K chiều có giá trị thực có tổng bằng 1. Giá trị nhập có thể dương, âm, bằng 0 hoặc lớn hơn 1, nhưng hàm *Softmax* sẽ luôn biến chúng thành một giá trị nằm trong khoảng (0:1]. Như vậy, chúng có thể được gọi là “**xác suất**”. Nếu một trong các giá trị nhập rất nhỏ hoặc âm, hàm *Softmax* biến chúng thành 1 xác suất nhỏ. Còn nếu một giá trị nhập lớn thì nó sẽ được chuyển thành một xác suất lớn. Nhưng xác suất luôn lớn hơn 0 và nhỏ hơn 1, hoặc bằng 1.

## PHẦN 3: THỰC NGHIỆM

### 1. BỘ DỮ LIỆU

Bộ dữ liệu chứa gần **3000 hình ảnh** về **8 loại** côn trùng phổ biến gây hại trong nông nghiệp. Các hình ảnh đa dạng về kích thước và độ phân giải được sưu tầm từ các nguồn khác nhau như: ảnh chụp, ảnh vẽ, ảnh tư liệu,... Bộ dữ liệu được lưu trữ và công bố tại *Kaggle* với tên chính thức là “*Dangerous Farm Insect*”. Chi tiết bộ dữ liệu bao gồm:

<i>Tên loài (Tiếng Việt)</i>	<i>Tên loài (Tiếng Anh)</i>	<i>Số lượng hình ảnh</i>
Bọ rùa 28 chấm	Epilachna vigintioctopunctata	328
Sâu xám	Cutworm	344
Bọ xít nâu dài	Riptortus	365
Sâu bắp cải	Cabbage Looper	324
Bọ xít hút máu	Boxelder Bugs	384
Ruồi Địa Trung Hải	Mediterranean Fruit Fly	394
Bọ xít	Squash Bug	358
Bọ nhảy	Flea Beetles	398
<b><i>Tổng cộng</i></b>		<b>2,895</b>

Toàn bộ hình ảnh trong bộ dữ liệu được chia ngẫu nhiên thành ba tập dữ liệu huấn luyện, đánh giá và kiểm thử với tỉ lệ **60/20/20**. Tức **60%** bộ dữ liệu được chia về *tập huấn luyện*, **20%** tiếp theo được chia về *tập đánh giá* và **20%** còn lại là *tập kiểm thử*.

### 2. ĐỘ ĐO

Độ đo được sử dụng để đánh giá mức độ hiệu quả của các phương pháp xuất trích đặc trưng và các mô hình máy học là *F1-score*. *F1-score* còn được gọi là *harmonic mean* của *precision* và *recall*. Công thức tính toán *F1-score*:

$$F1 = 2 \frac{Precision * Recall}{Precision + Recall}$$

Cụ thể, *Macro F1-score* được sử dụng là độ đo chính trong toàn bộ đồ án vì đây là bài toán phân loại đa lớp (**8 phân lớp**). *Macro F1-score* được tính bằng cách lấy trung bình *F1-score* của tất cả các lớp. Ngoài ra, *F1-score* còn cho phép đánh giá độ hiệu quả của mô hình trên cả độ chính xác (*Precision*) và tỉ lệ bỏ sót (*Recall*). Vì thế, mục tiêu là nhằm chọn ra mô hình có *F1-score* càng cao càng tốt, khi đó mô hình có độ chính xác cao và tỉ lệ bỏ sót thấp.

Ngoài ra, nhằm tìm ra tổ hợp các siêu tham số cho ra độ hiệu quả cao nhất cho từng phương pháp xuất trích đặc trưng và mô hình máy học, *Macro F1-score* được tổng hợp lại dưới dạng *Macro F1-score lớn nhất (Max Macro F1-score)*.

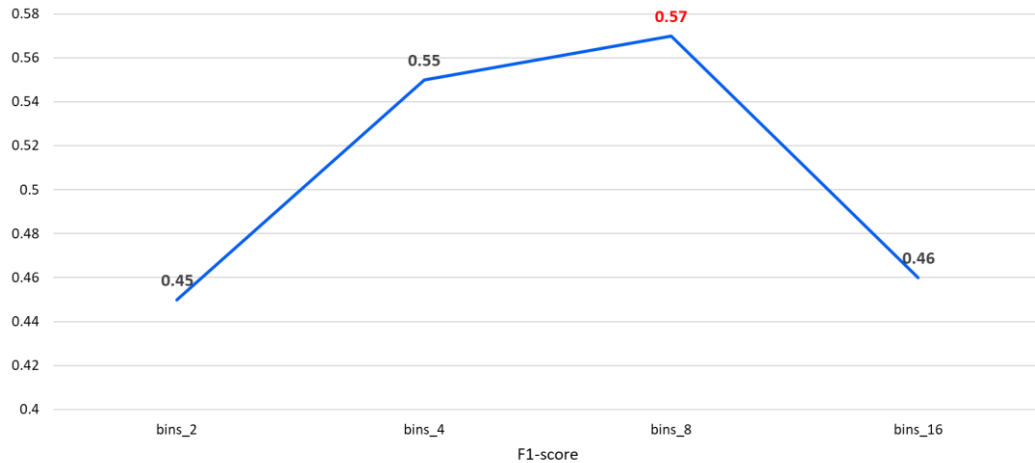
### 3. KẾT QUẢ

#### 3.1. So sánh tham số bins của phương pháp HSV Histogram

Mô hình máy học *SVM* với bộ siêu tham số mặc định được sử dụng nhằm tìm ra tham số *bins* của phương pháp xuất trích đặc trưng *HSV Histogram* có *Macro F1-score* cao nhất. Bảng dưới đây là các giá trị chi tiết áp dụng cho tham số *bins* trên:

<i>Tham số</i>	<i>Ý nghĩa</i>	<i>Giá trị</i>
bins	Số lượng khoảng giá trị được dùng chia các kênh màu	2, 4, 8, 16

Biểu đồ thể hiện quá trình thực nghiệm các trường hợp *bins* của phương pháp *HSV Histogram* với mô hình máy học *SVM* cùng bộ siêu tham số được tối ưu riêng cho từng *bins*. Quá trình so sánh tham số *bins* được thực hiện dựa trên hệ đo lường *Macro F1-score*:



Qua thực nghiệm, phương pháp *HSV Histogram* đạt *Macro F1-score* cao nhất (**0.57**) khi tham số *bins* bằng 8.

Giá trị các bộ siêu tham số ứng với các *bins* của phương pháp *HSV Histogram* 2, 4, 8, 16:

<i>bins</i>	<i>C</i>	<i>kernel</i>	<i>degree</i>	<i>gamma</i>	<i>F1-score</i>
2	90.6	poly	20	scale	0.45
4	29.4	poly	20	scale	0.55
8	90	poly	10	scale	0.57
16	27	poly	20	scale	0.46

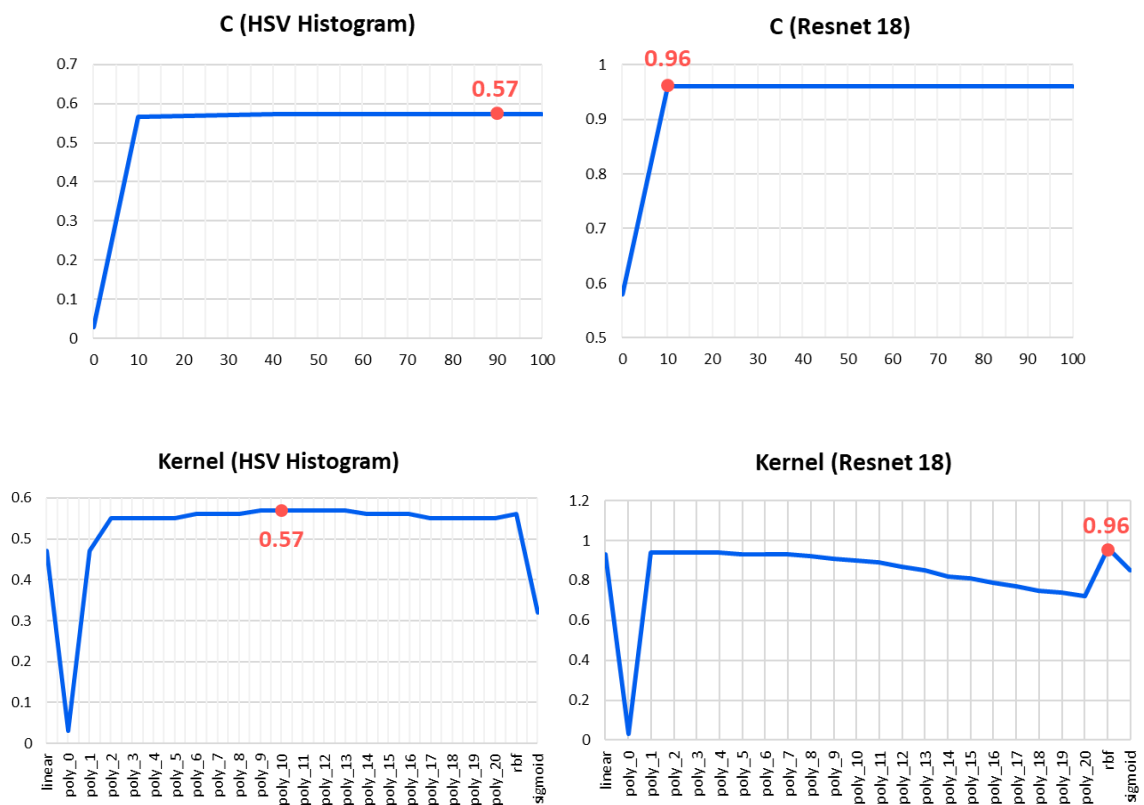
### 3.2. So sánh siêu tham số mô hình SVM bằng phương pháp Grid Search

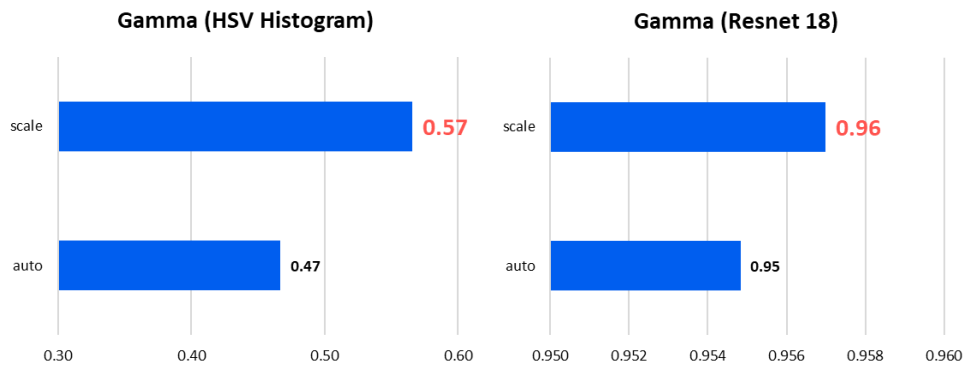
Phương pháp *Grid Search* được sử dụng để thử qua tất cả tổ hợp siêu tham số của mô hình máy học *SVM* nhằm tìm ra tổ hợp có độ chính xác cao nhất. Các siêu tham số (*C*, *kernel*, *degree*, *gamma*) được điều chỉnh bằng phương pháp *Grid Search* vì có ảnh hưởng lớn đến độ hiệu quả của mô hình *SVM*. Ngoài ra, các siêu tham số còn lại được cài đặt mặc định theo đề xuất của *Scikit-Learn*. Bảng dưới đây là các khoảng giá trị chi tiết áp dụng cho các siêu tham số trên:

<i>Siêu tham số</i>	<i>Ý nghĩa</i>	<i>Khoảng giá trị</i>	<i>Bước nhảy</i>
C	Tham số điều chỉnh L2 Regularisation (Tỉ lệ	(0, 100]	10/5/1/0.1

	ngược với độ lớn Regulariation)		
kernel	Thủ thuật ánh xạ kernel	linear, poly, rbf, sigmoid	
degree	Chiều không gian dữ liệu sử dụng cho thủ thuật kernel polynomial	[0, 20]	1
gamma	Hệ số kernel áp dụng cho thủ thuật kernel (rbf, poly, sigmoid)	scale, auto	

Các biểu đồ thể hiện quá trình thực nghiệm phương pháp *Grid Search* với mô hình máy học *SVM* cùng *tập dữ liệu đánh giá*. Quá trình so sánh các siêu tham số được thực hiện dựa trên hệ đo lường *Macro F1-score cao nhất*:





Qua thực nghiệm bằng phương pháp *Grid Search*, *SVM* đạt được độ chính xác cao nhất lần lượt với đặc trưng *HSV Histogram* là **57%** và đặc trưng *ResNet18* là **96%**.

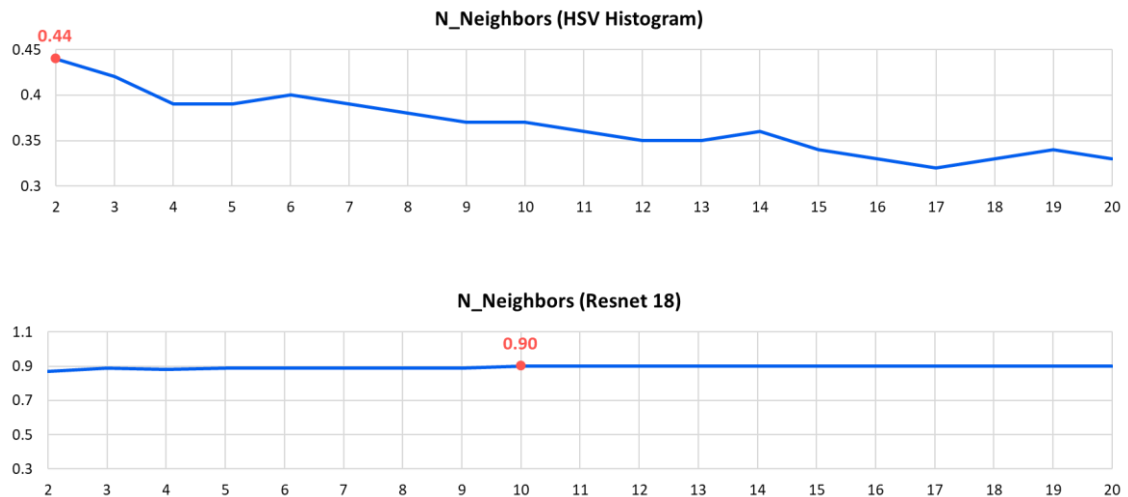
	<i>HSV Histogram</i>	<i>ResNet18</i>
C	90	10
kernel	poly	rbf
degree	10	—
gamma	scale	scale
<b>Độ chính xác cao nhất</b>	<b>0.57</b>	<b>0.96</b>

### 3.3. So sánh siêu tham số mô hình KNN bằng phương pháp Grid Search

Phương pháp *Grid Search* được sử dụng để thử qua tất cả trường hợp siêu tham số  $n\_neighbors$  của mô hình máy học *KNN* nhằm tìm ra trường hợp có độ chính xác cao nhất. Ngoài ra, các siêu tham số còn lại được cài đặt mặc định theo đề xuất của *Scikit-Learn*. Bảng dưới đây là khoảng giá trị chi tiết áp dụng cho siêu tham số  $n\_neighbors$  trên:

<i>Siêu tham số</i>	<i>Ý nghĩa</i>	<i>Khoảng giá trị</i>	<i>Bước nhảy</i>
$n\_neighbors$	số lượng điểm dữ liệu lân cận mô hình sử dụng để phân loại	[2, 20]	1

Biểu đồ thể hiện quá trình thực nghiệm phương pháp *Grid Search* với mô hình máy học *KNN* cùng *tập dữ liệu đánh giá*. Quá trình so sánh siêu tham số được thực hiện dựa trên hệ đo lường *Macro F1-score cao nhất*:



Qua thực nghiệm bằng phương pháp *Grid Search*, *KNN* đạt được độ chính xác cao nhất lần lượt với đặc trưng *HSV Histogram* là **44%** và đặc trưng *ResNet18* là **90%**.

	<i>HSV Histogram</i>	<i>ResNet18</i>
n_neighbors	2	10
<b>Độ chính xác cao nhất</b>	<b>0.44</b>	<b>0.90</b>

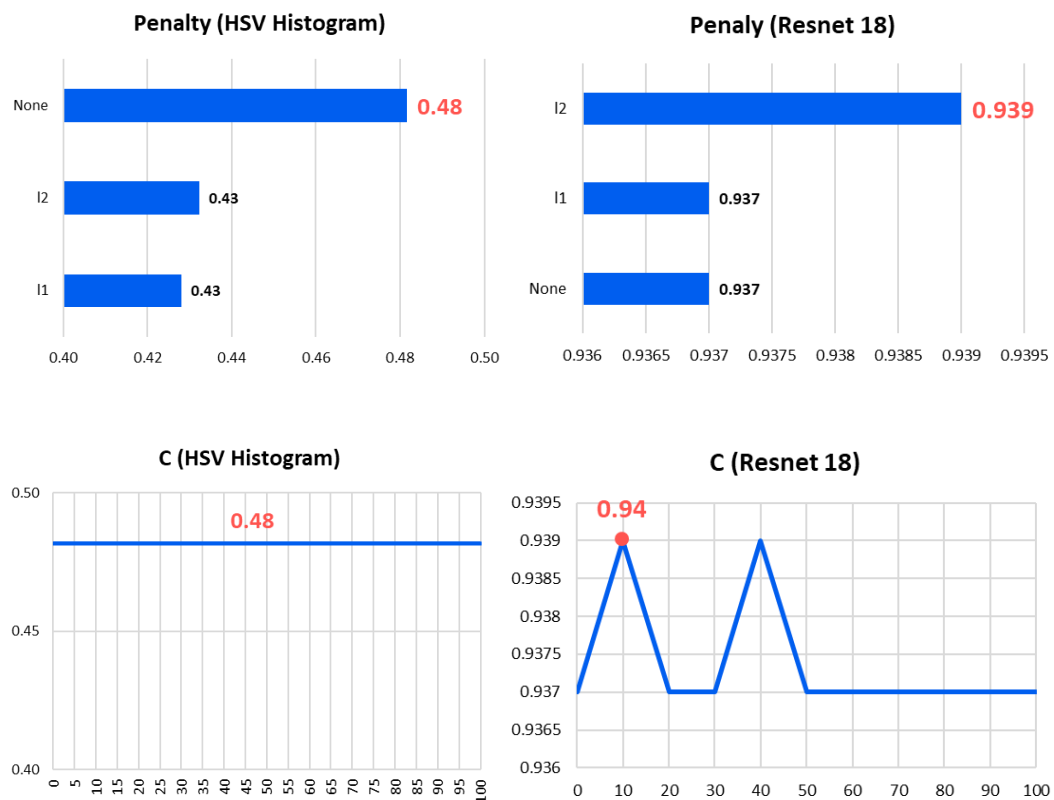
### 3.4. So sánh siêu tham số mô hình *Softmax Regression* bằng phương pháp *Grid Search*

Phương pháp *Grid Search* được sử dụng để thử qua tất cả tổ hợp siêu tham số của mô hình máy học *Softmax Regression* nhằm tìm ra tổ hợp có độ chính xác cao nhất. Các siêu tham số (*penalty*, *C*, *solver*, *multi\_class*) được điều chỉnh bằng phương pháp *Grid Search* vì có ảnh hưởng lớn đến độ hiệu quả của mô hình *Softmax Regression*. Ngoài ra, các siêu tham số còn lại được cài đặt mặc định theo đề xuất của *Scikit-Learn*. Bảng dưới đây là các khoảng giá trị chi tiết áp dụng cho các siêu tham số trên:

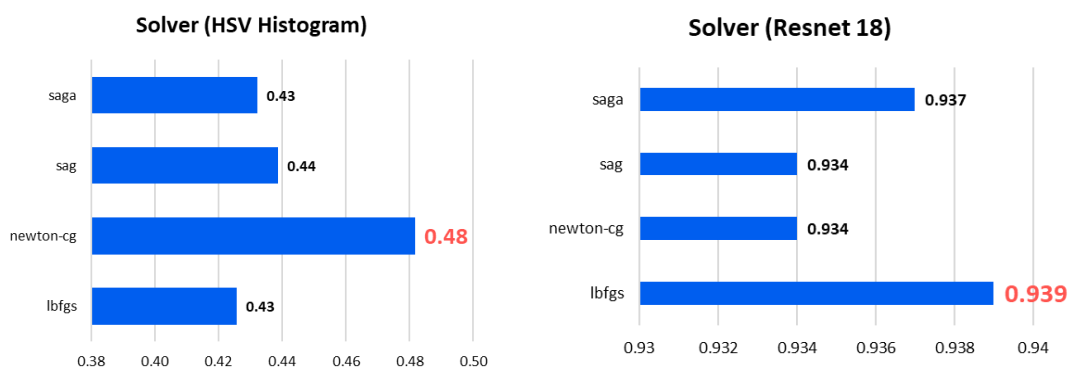
<i>Siêu tham số</i>	<i>Ý nghĩa</i>	<i>Khoảng giá trị</i>	<i>Bước nhảy</i>
---------------------	----------------	-----------------------	------------------

penalty	Thuật toán Regulariation	l1, l2, None	
C	Tham số điều chỉnh L2 Regulariation (Tỉ lệ nghịch với độ lớn Regulariation)	(0, 100]	5/1/0.1
solver	Thuật toán tối ưu	lbfgs, newton-cg, sag, saga	
multi_class	Điều chỉnh mô hình Logistic Regression sang Softmax Regression	multinomial	

Các biểu đồ thể hiện quá trình thực nghiệm phương pháp *Grid Search* với mô hình máy học *Softmax Regression* cùng *tập dữ liệu đánh giá*. Quá trình so sánh các siêu tham số được thực hiện dựa trên hệ đo lường *Macro F1-score cao nhất*:







Qua thực nghiệm bằng phương pháp *Grid Search*, *Softmax Regression* đạt được độ chính xác cao nhất lần lượt với đặc trưng *HSV Histogram* là **48%** và đặc trưng *ResNet18* là trên **94%**.

	<i>HSV Histogram</i>	<i>ResNet 18</i>
penalty	None	l2
C	—	10
solver	Newton-cg	lbfgs
multi_class	multinomial	multinomial
<b>Độ chính xác cao nhất</b>	<b>0.48</b>	<b>0.94</b>

### 3.5. So sánh kết quả

Qua quá trình thực nghiệm bằng phương pháp *Grid Search* như trên, đặc trưng *ResNet18* có *Macro F1-score* cao gấp đôi so với đặc trưng *HSV Histogram* (~44%). Ở cả 2 đặc trưng, *SVM* luôn đem lại kết quả cao nhất (57% / 96%) và *KNN* luôn cho kết quả thấp nhất (44% / 90%). Tập dữ liệu kiểm thử và hệ đo lường *Macro F1-score* được sử dụng cho quá trình đánh giá cuối cùng:

<i>Độ chính xác</i>	<i>HSV Histogram</i>	<i>ResNet18</i>	<i>Chênh lệch</i>
SVM	0.57	0.96	+0.39
KNN	0.44	0.90	+0.46

Softmax Regression	0.48	0.94	+0.46
<i>Trung Bình</i>	0.50	0.92	<b>+0.44</b>

## PHẦN 4: ĐÁNH GIÁ

### 1. ƯU VÀ NHƯỢC ĐIỂM CỦA ĐẶC TRƯNG HSV HISTOGRAM

#### 1.1. Ưu điểm

**Đơn giản và dễ hiểu:** Histogram tính toán nhanh, đơn giản, thích hợp trong các ứng dụng thời gian thực.

**Bất biến với phép quay:** Histogram là rotation invariant, vì khi một ảnh bị xoay đi thì phân phối màu hầu như là không đổi.

#### 1.2. Nhược điểm

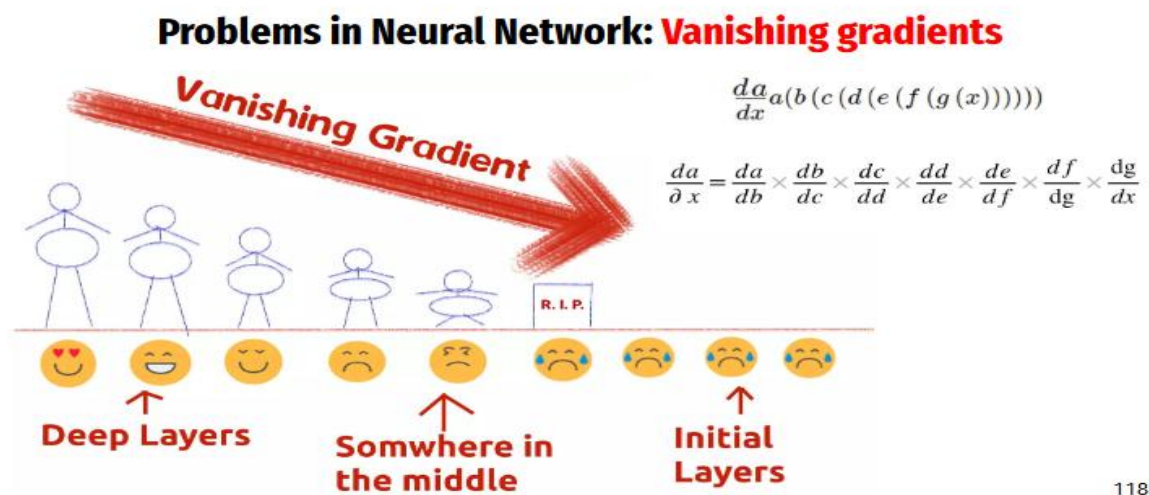
**Mất thông tin về hình dáng, cấu trúc:** Các ảnh dù cho rất khác nhau nhưng vẫn có phân phối màu giống nhau – trường hợp bàn cờ vua và ảnh nửa trắng, nửa đen.

**Phụ thuộc vào kích thước và số lượng khoảng (bins):** Có thể bị ảnh hưởng lớn bởi cách chia khoảng (bins). Việc chọn số lượng khoảng không hợp lý có thể dẫn đến mất thông tin hoặc gây nhầm lẫn trong phân tích.

### 2. ƯU VÀ NHƯỢC ĐIỂM CỦA ĐẶC TRƯNG RESNET

#### 2.1. Ưu điểm

**Giải quyết vấn đề vanishing gradient:** *ResNet* sử dụng các kỹ thuật “skip connection” và “identity mapping” giúp truyền tải thông tin hiệu quả hơn qua mạng, giảm thiểu ảnh hưởng của vanishing gradient, từ đó tăng tốc độ huấn luyện và cải thiện độ chính xác.



**Đạt độ chính xác cao:** Trên các benchmark tiêu chuẩn như *ImageNet*, *CIFAR-100* và *COCO*, *ResNet* ghi nhận hiệu suất vượt trội so với các mô hình CNN trước đây.

**Hiệu suất tính toán tối ưu:** Nhờ vào thiết kế thông minh của mình, *ResNet* tối ưu quá trình truyền tải thông tin giúp giảm thiểu chi phí tính toán và tiết kiệm thời gian đào tạo. So với các mô hình CNN khác, *ResNet* có thể đạt được độ chính xác cao với lượng tài nguyên tính toán thấp hơn.

**Khả năng mở rộng:** *ResNet* có thể được mở rộng thành các mô hình lớn hơn như *ResNet50* và *ResNet101*, từ đó đạt được độ chính xác cao hơn.

## 2.2. Nhược điểm

**Kích thước mô hình lớn:** So với các mô hình CNN đơn giản hơn, *ResNet* có kích thước mô hình lớn hơn, dẫn đến yêu cầu cao về dung lượng lưu trữ và tài nguyên tính toán. Điều này có thể gây khó khăn trong việc triển khai *ResNet* trên các thiết bị có cấu hình thấp hoặc trong các ứng dụng có hạn chế về tài nguyên.

**Yêu cầu dữ liệu đào tạo lớn:** Để đạt được độ chính xác cao nhất, *ResNet* cần được đào tạo với một lượng lớn dữ liệu. Việc thu thập và xử lý dữ liệu này có thể tốn thời gian và chi phí. Đây là một hạn chế đáng kể đối với các dự án có dữ liệu hạn chế hoặc ngân sách eo hẹp.

**Nguy cơ overfitting:** Với kích thước mô hình lớn và khả năng học tập mạnh mẽ, *ResNet* có nguy cơ overfitting cao, đặc biệt khi được đào tạo với tập dữ liệu nhỏ. Việc

áp dụng các kỹ thuật chống overfitting là cần thiết để đảm bảo hiệu quả của mô hình trên dữ liệu chưa từng gặp.

## PHẦN 4: TỔNG KẾT

### 1. BÀI TOÁN

Dựa trên kết quả thực nghiệm so sánh giữa trích xuất đặc trưng của *Resnet18* và *HSV Histogram* trong việc phân loại loài côn trùng gây hại đã cho thấy sự khác biệt rõ rệt về độ chính xác. Đặc trưng *Resnet18* đã thể hiện sự ưu việt với độ chính xác gấp đôi so với *HSV Histogram*. Đồng thời, việc sử dụng *SVM (Support Vector Machine)* luôn mang lại kết quả cao nhất với độ chính xác lên tới **95%** khi kết hợp với đặc trưng *Resnet18* và **61%** khi kết hợp với *HSV Histogram*. Trong khi đó phương pháp *KNN (K-Nearest Neighbors)* luôn cho kết quả thấp nhất với độ chính xác **90%** khi sử dụng đặc trưng *Resnet18* và chỉ khoảng **48%** khi sử dụng với đặc trưng *HSV Histogram*. Tóm lại, việc kết hợp đặc trưng *Resnet18* với *SVM* không chỉ là lựa chọn tối ưu nhất mà còn mang lại độ chính xác cao nhất và hiệu suất ấn tượng trong việc phân loại côn trùng gây hại.

### 2. MỨC ĐỘ HOÀN THÀNH

Công việc	225203390	22520189	22520084
Giới thiệu bài toán	X	X	X
Phương pháp thực hiện	X	X	X
Thực nghiệm	X	X	X
Đánh giá & Kết luận	X	X	X

<b>Mức độ hoàn thành các công việc được phân công (%)</b>	100	100	100
---	-----	-----	-----

## PHẦN 5: TÀI LIỆU THAM KHẢO

### 1. Notebook:

<https://colab.research.google.com/drive/1l6zGxIrJCcsH5UQsMal5efVE8em31KIo>

### 2. Dataset:

<https://www.kaggle.com/datasets/snnasng/dangerous-farm-insect>

3. <https://viblo.asia/p/huong-dan-tat-tan-tat-ve-pytorch-de-lam-cac-bai-toan-ve-ai-YWOZrNkNZQ0>
4. <https://viblo.asia/p/gioi-thieu-mang-resnet-vyDZOa7R5wj>
5. <https://machinelearningcoban.com/2017/04/09/smv/>
6. <https://machinelearningcoban.com/2017/01/08/knn/>
7. <https://machinelearningcoban.com/2017/02/17/softmax/>
8. [https://docs.opencv.org/4.x/d1/db7/tutorial\\_py\\_histogram\\_begins.html](https://docs.opencv.org/4.x/d1/db7/tutorial_py_histogram_begins.html)