IMAGE CLASSIFICATION VỚI HOG VÀ SVM

Lập trình song song ứng dụng
Nhóm: Parallel Pioneers

Thành viên

Đỗ Lê Khánh Đăng - 19120186
Nguyễn Đắc Thắng - 19120364
Lục Minh Bửu - 19120462

Nội dung

- Giới thiệu đồ án
- Mục tiêu đồ án
- Thuật toán HOG
- Song song HOG
- Cải tiến HOG
- Thuật toán SVM
- Cải tiến SVM - Accuracy
- Cải tiến SVM - song song hóa
- So sánh và đánh giá

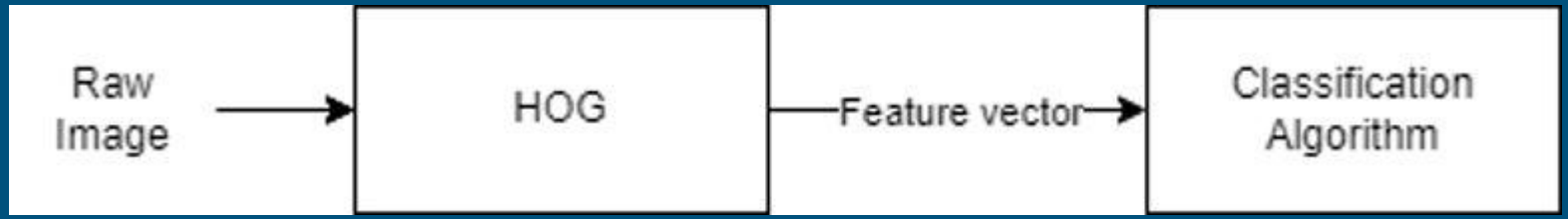
Giới thiệu đồ án

Chủ đề: Image classification với HOG và SVM.

Khi bộ dữ liệu không đủ lớn, CNN hay các mạng học sâu tỏ ra kém hiệu quả và tốn nhiều tài nguyên để train mô hình. HOG và SVM được cho rằng sẽ khắc phục được điểm yếu của CNN trong trường hợp này với các ưu điểm:

- Tốc độ nhanh hơn
- Kết quả tốt hơn
- Ít tiêu tốn tài nguyên hơn

Sơ lược về mô hình HOG và SVM



Mục tiêu đồ án ban đầu

- Mức 100%: hoàn thành được thuật toán HOG phiên bản tuần tự, song song và các cải tiến với độ chính xác cao và tối ưu thời gian thực thi, có thể áp dụng vào các bài toán realtime
- Mức 125%: hoàn thành toàn bộ hệ thống nhận diện gồm HOG + SVM phiên bản tuần tự, song song hóa và thực hiện các cải tiến để hệ thống có tốc độ nhanh hơn hoặc bằng thư viện, với độ chính xác ít nhất là xấp xỉ thư viện.
- Mức 75%: cài đặt được HOG tuần tự và song song

Thuật toán HOG

- Làm xám ảnh
- Tính Gradients theo hướng ngang và dọc với hai ma trận: [-1 0 1] * I và [-1 0 1]^T * I
- Tính mức độ thay đổi (magnitude) và hướng thay đổi (direction) của Gradient.
- Tìm histogram của gradient cell.
- Chuẩn hóa block.
- Tính vector đặc trưng HOG.

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

Đây là công việc tiêu tốn nhiều tài nguyên tính toán và có thể song song hóa.

Thuật toán HOG

Làm xám ảnh

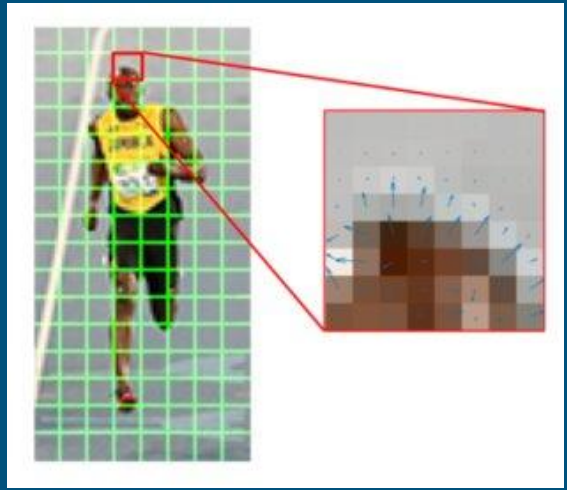
Tính G_x , G_y (convolution)

Tính magnitude, magnitude

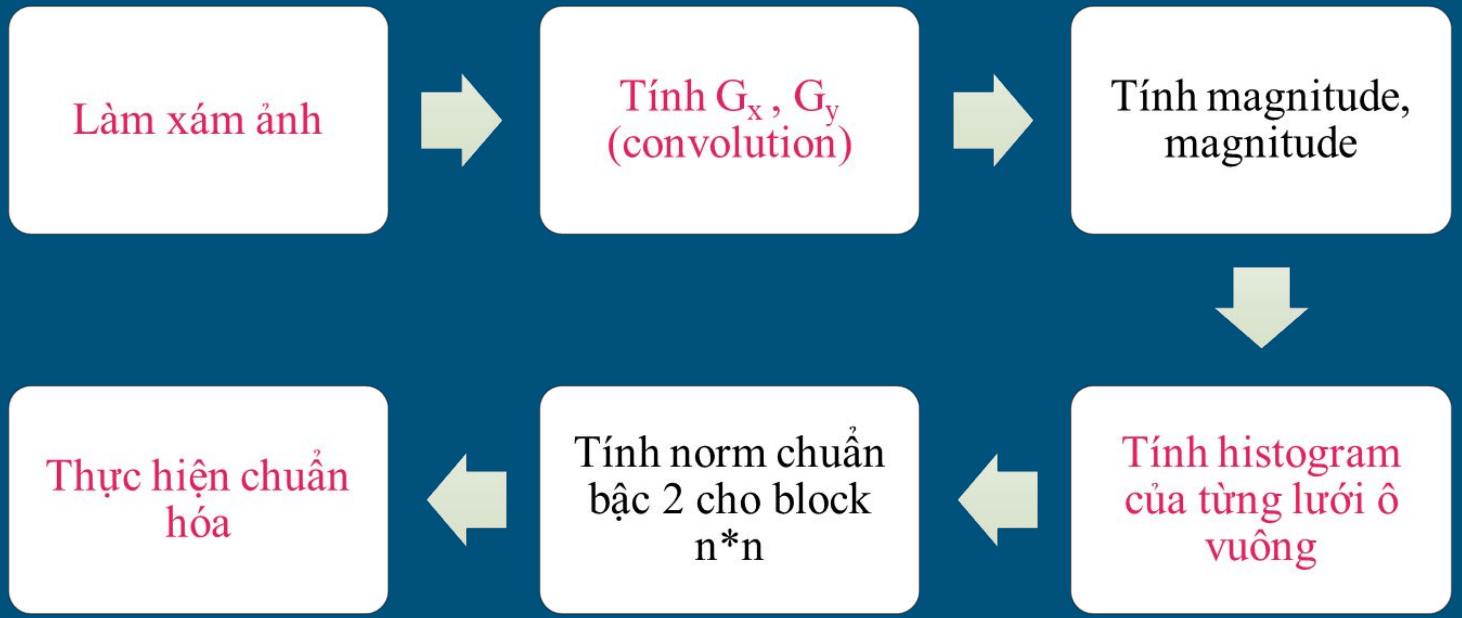
Tính histogram của từng lưới ô vuông

Tính norm chuẩn bậc 2 cho block n*n

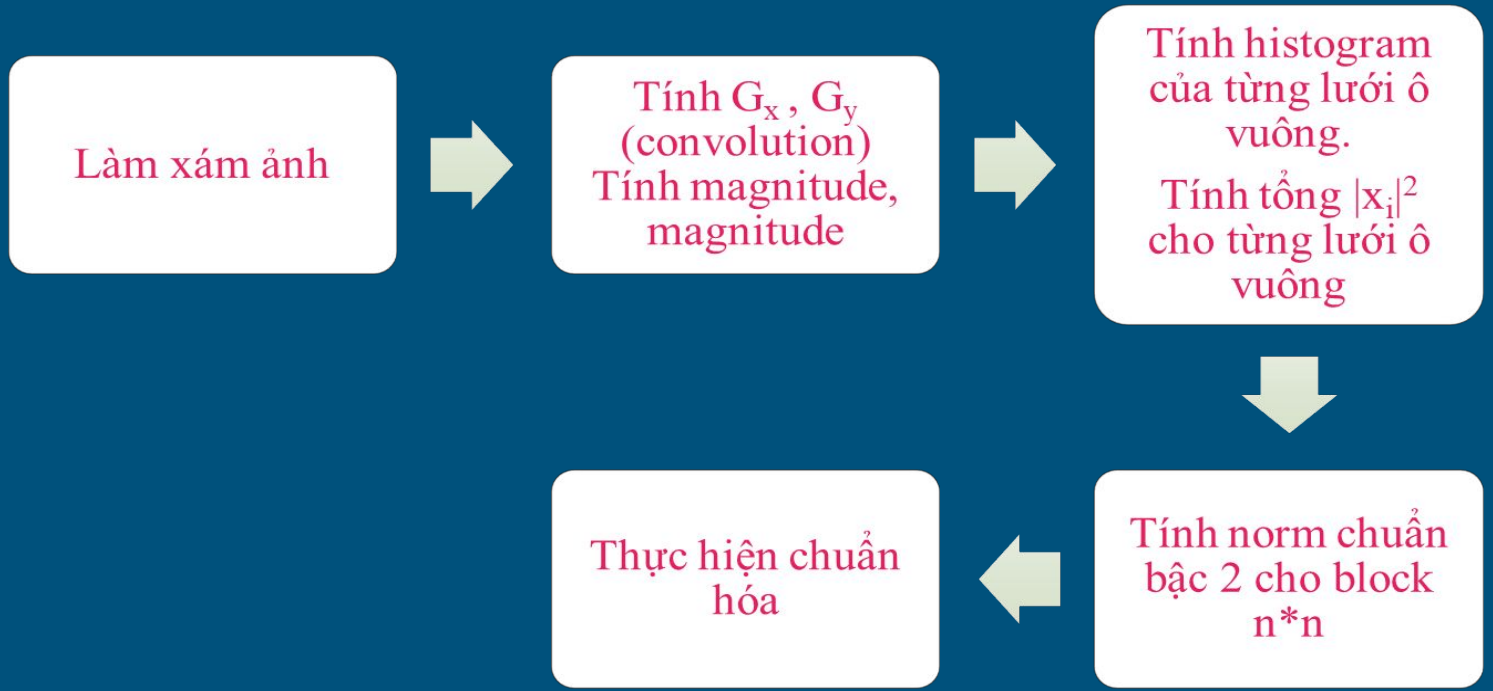
Thực hiện chuẩn hóa



Thuật toán HOG - version 1



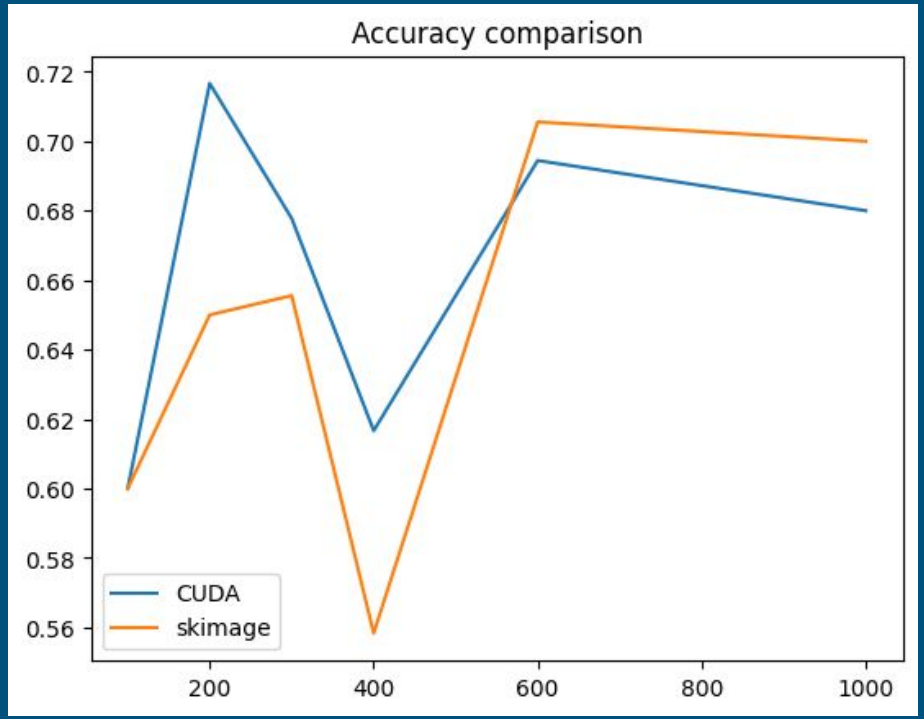
Thuật toán HOG - version 2

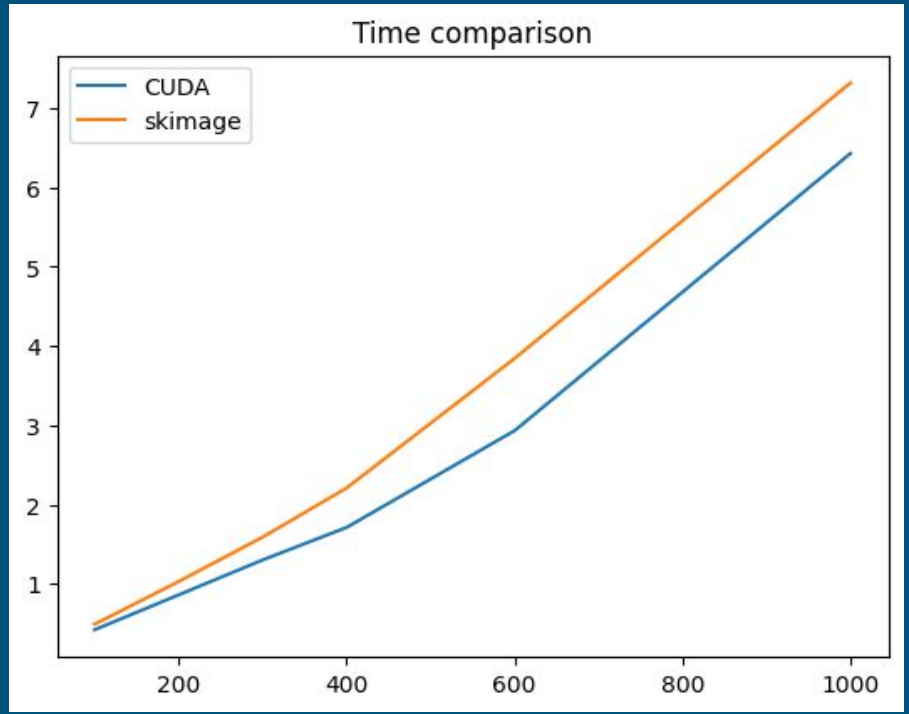


So sánh thời gian các phiên bản HOG

Phiên bản	Thời gian chạy
Tuần tự	18.5 s ± 309 ms per loop
Thư viện	987 ms ± 42 ms per loop
Song song – Ver 1	584 ms ± 24.7 ms per loop
Song song – Ver 2	34.2 ms ± 1.66 ms per loop

So sánh với thư viện skimage

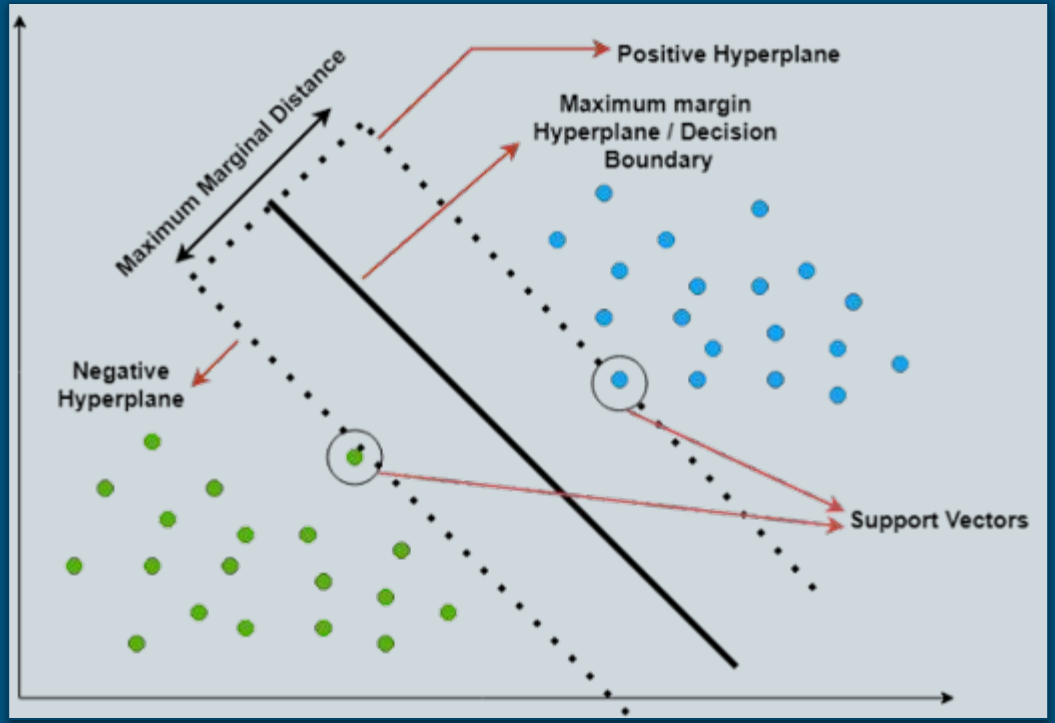




SVM cũ

Sử dụng linear kernel và tối ưu bằng Gradient Descent

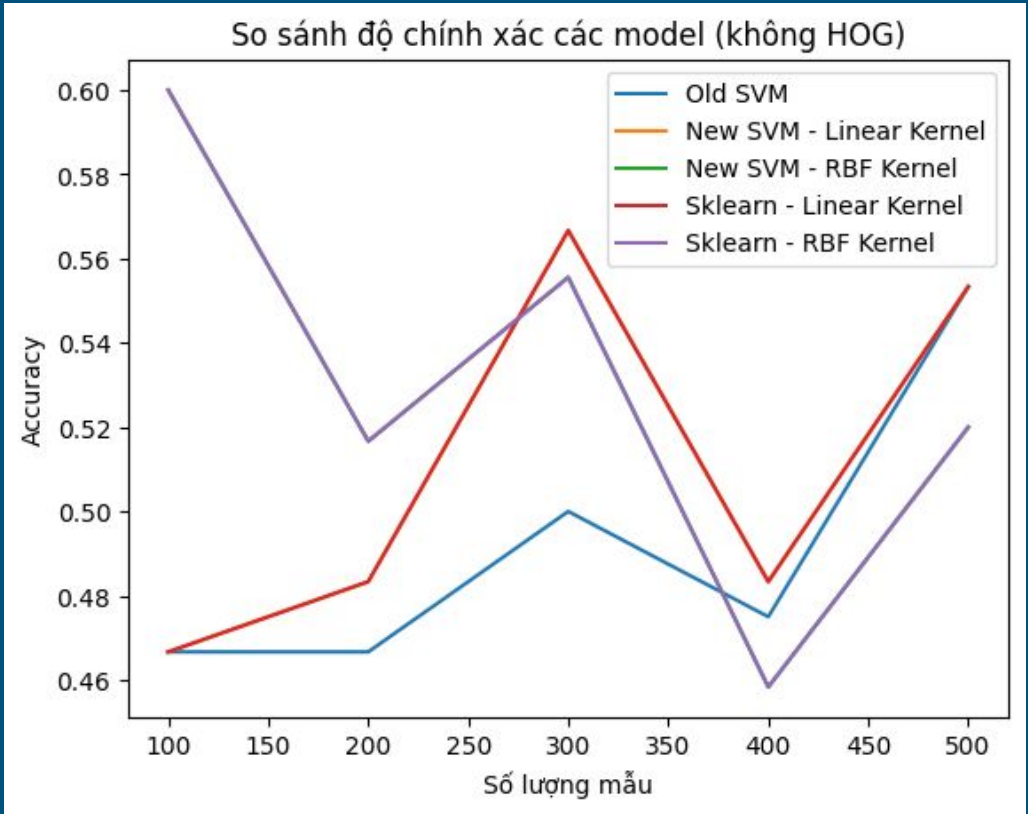
Nguồn: https://www.pycodemates.com/2022/07/support-vector-machines-detailed-overview.html



SVM mới

- Những thay đổi:
  - SVM cũ: sử dụng kernel linear, tối ưu bằng Gradient Descent
  - SVM mới: 2 option cho kernel linear hoặc rbf, tối ưu bằng SMO
- Lý do:
  - RBF biểu diễn mối quan hệ phi tuyến của các điểm dữ liệu
  - Kernel RBF khi số lượng dữ liệu đủ lớn cho kết quả tốt hơn linear
  - Khi áp dụng RBF thì Gradient Descent kém hiệu quả
  - SMO được thư viện sklearn áp dụng
  - SMO là thuật toán được thiết kế dành cho SVM

So sánh SVM mới và cũ không HOG



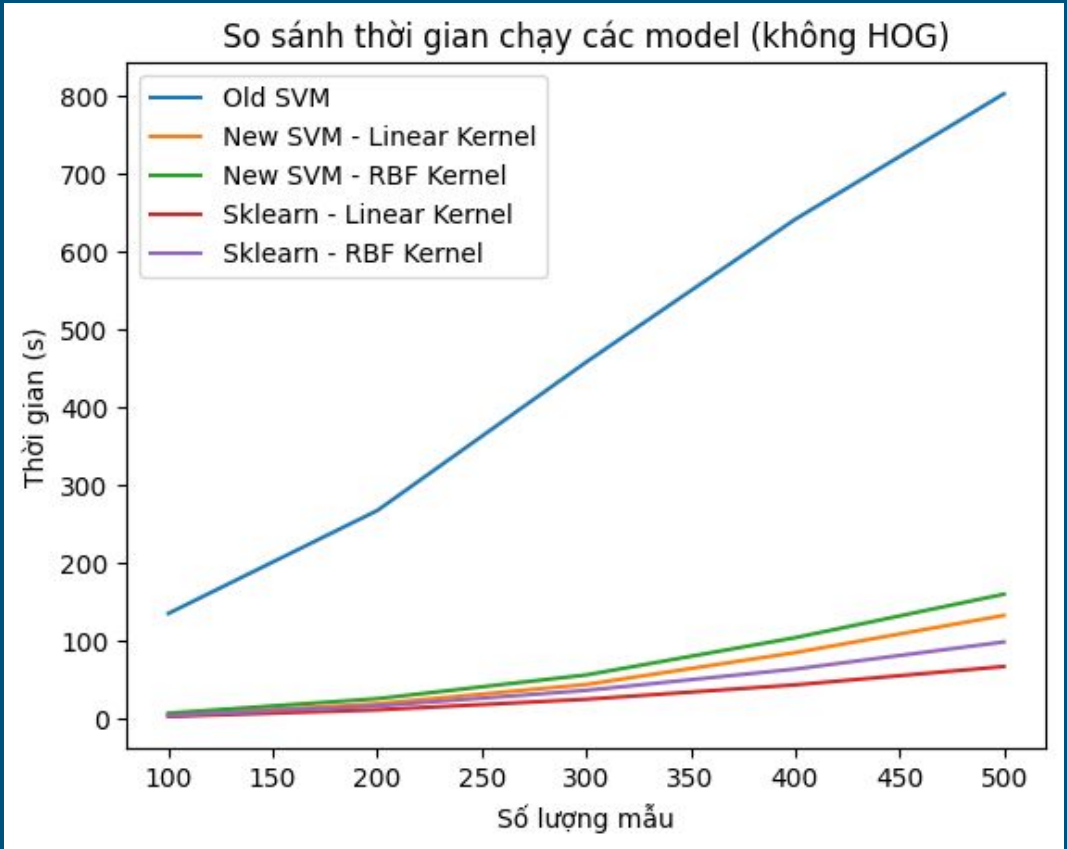
- SVM bản cũ có accuracy thấp hơn trong mọi trường hợp
- SVM mới có accuracy giống hoàn toàn với thư viện sklearn.
- Kernel RBF kém ổn định hơn và thấp hơn linear khi số lượng mẫu thấp.

So sánh SVM mới và cũ không HOG

- Chi tiết số liệu:

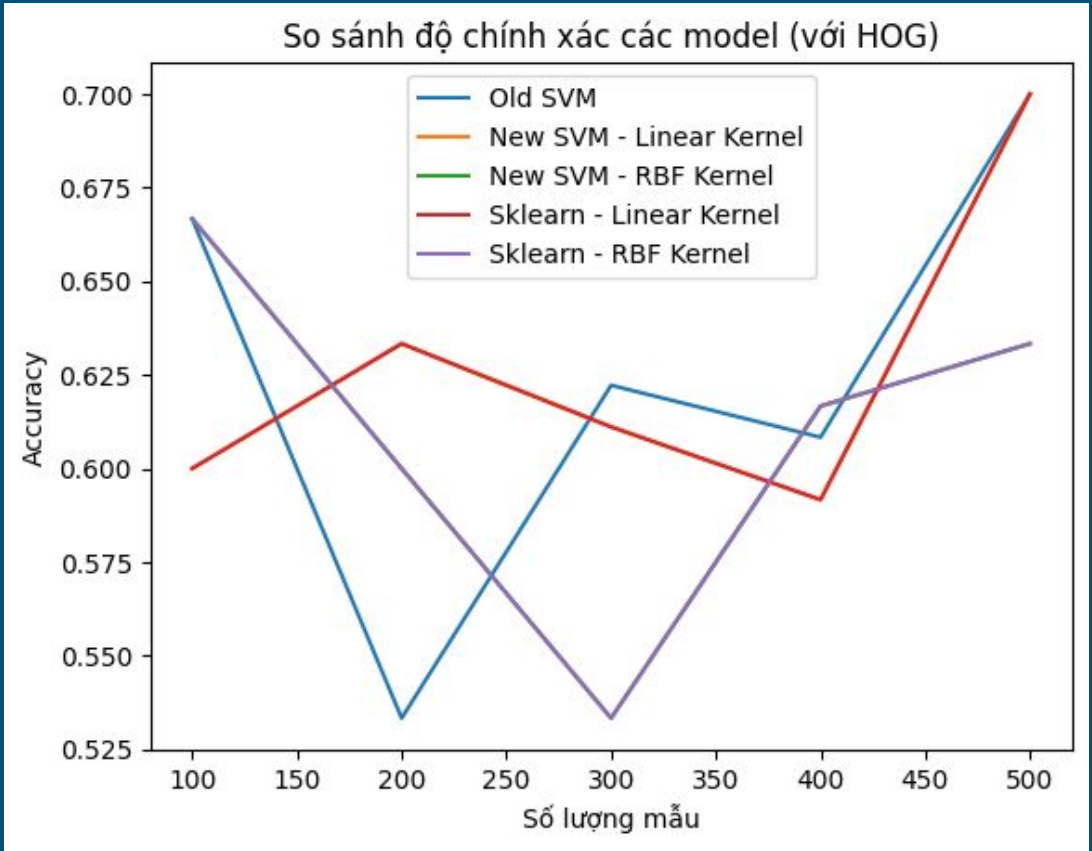
	Số lượng mẫu	Old SVM Acc	New SVM Linear Acc	New SVM RBF Acc	Sklearn Linear Acc	Sklearn RBF Acc
0	100	0.466667	0.466667	0.600000	0.466667	0.600000
1	200	0.466667	0.483333	0.516667	0.483333	0.516667
2	300	0.500000	0.566667	0.555556	0.566667	0.555556
3	400	0.475000	0.483333	0.458333	0.483333	0.458333
4	500	0.553333	0.553333	0.520000	0.553333	0.520000

So sánh SVM mới và cũ không HOG



- SVM cũ có thời gian chạy lâu nhất.
- Sklearn có thời gian chạy thấp nhất, nhưng không quá khác biệt so với SVM mới

So sánh SVM mới và cũ có HOG



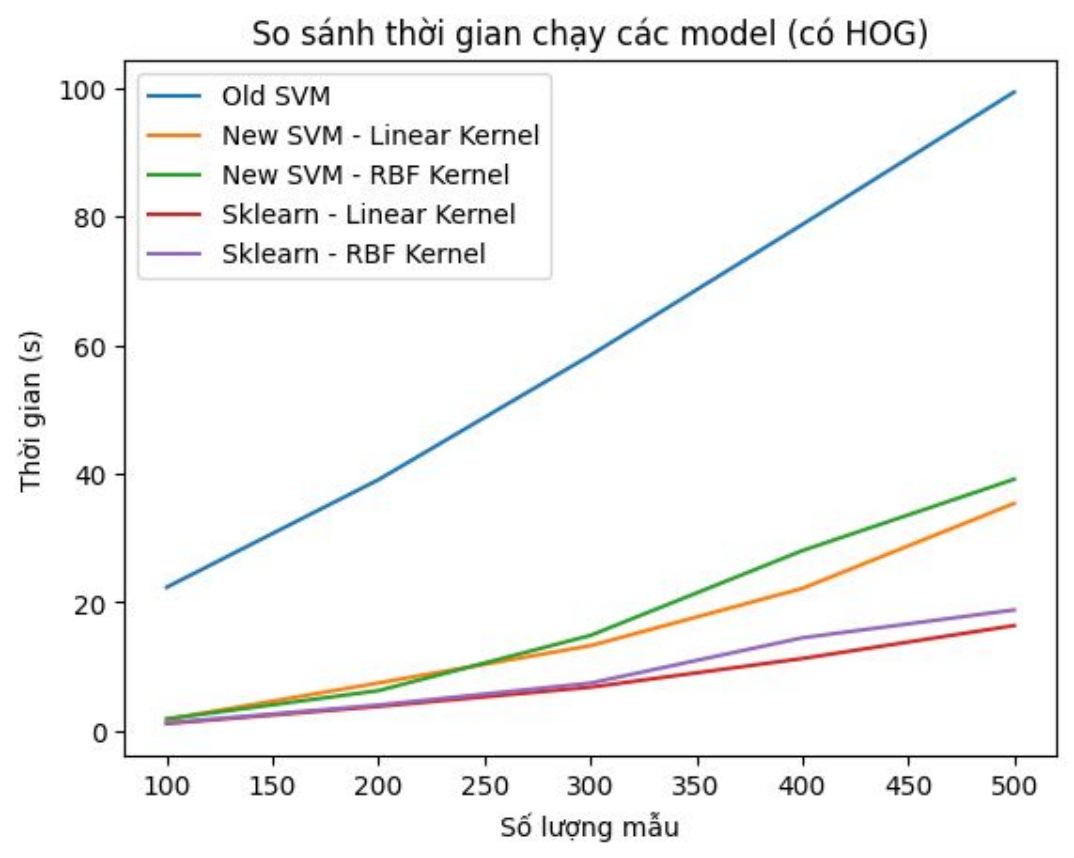
- SVM bản cũ có accuracy thấp hơn trong đa số trường hợp
- SVM mới có accuracy giống hoàn toàn với thư viện sklearn.

So sánh SVM mới và cũ có HOG

Số liệu chi tiết về Accuracy:

Số lượng mẫu	Old SVM + hog Acc	New SVM Linear + hog Acc	New SVM RBF + hog Acc	Sklearn Linear + hog Acc	Sklearn RBF + hog Acc
100	0.666667	0.600000	0.666667	0.600000	0.666667
200	0.533333	0.633333	0.600000	0.633333	0.600000
300	0.622222	0.611111	0.533333	0.611111	0.533333
400	0.608333	0.591667	0.616667	0.591667	0.616667
500	0.700000	0.700000	0.633333	0.700000	0.633333

So sánh SVM mới và cũ có HOG



- SVM bản mới chạy nhanh hơn bản cũ (tính cả predict và fit).
- SVM của sklearn chạy nhanh nhất nhưng không quá nhanh so với SVM bản mới.

Song song hóa SVM mới

- Khó khăn:
  - SMO (Sequential Minimal Optimization) bản chất là thuật toán tuần tự
  - Hầu như không phân thành các thao tác độc lập để song song.
- Giải quyết:
  - Phân thuật toán thành 3 giai đoạn lớn và đo tổng thời gian chạy: xác định cặp điểm cần tối ưu, tối ưu cặp điểm, cập nhật Gradient
  -
  - Thuật toán cần gọi hàm tính kernel K(i, j) và tính giá trị Qij = yi * yj * K(i, j) rất nhiều lần → Tính tổng thời gian mà thao tác này chiếm.

Song song hóa SVM mới

	100 mẫu	200 mẫu	300 mẫu
Thời gian fit	3.44s	14.54s	34.05s
Tìm cặp alpha	0.99s	4.07s	9.25s
Tối ưu alpha	0.0009s	0.0018s	0.0029s
Cập nhật Gradient	2.3s	10.16s	24.33s
Thời gian tính Q	3.2s	13.88s	32.06s
Thời gian predict	2.6s	11.72s	30.7s

Song song hóa SVM mới

- Thời gian tính Q chiếm đa số thời lượng fit, dù thao tác tính Qij không mất nhiều thời gian nhưng lại được gọi rất nhiều lần.
- → Tìm toàn bộ Q bằng song song và lưu vào 1 matrix. Khi cần chỉ cần lấy ra.
- Predict tốn nhiều thời gian bởi phải tính Kij nhiều lần (tương tự như tính Q ở fit). → Giải quyết tương tự
- Sau khi tính Q toàn bộ, có thể cập nhật Gradient nhanh hơn bằng các thao tác với ma trận thay vì vòng for.
- Ngoài ra có thể tính dual_coef=y*alpha bằng song song thay vì thực hiện y_i*alpha_i khi predict.

Song song hóa SVM mới

	100 mẫu	200 mẫu	300 mẫu
Thời gian fit	3.44s → 1.34s	14.54s → 1.67s	34.05s →4.26s
Tìm cặp alpha	0.99s → 0.06s	4.07s → 0.16s	9.25s → 0.33s
Tối ưu alpha	0.0009s → 0.0004s	0.0018s → 0.0006s	0.0029s → 0.0009s
Cập nhật Gradient	2.3s → 0.0008s	10.16s → 0.0012s	24.33s → 0.0017s
Thời gian tính Q	3.2s → 1s	13.88s → 1.19s	32.06s → 3.48s
Thời gian predict	2.6s →1.4s	11.72s → 1.03s	30.7s → 1.77s

Song song hóa SVM mới

- Ưu điểm:
  - Tối ưu hóa hiệu quả tổng thời gian fit và predict.
  - Không có nguy cơ mất mát thông tin như cơ chế shrink của sklearn (cơ chế giúp tăng tốc khi lượng data lớn) nhưng vẫn có tốc độ cao hơn sklearn.
- Nhược điểm:
  - Tốn nhiều VRAM vì phải load toàn bộ data lên GPU để tìm Q
  - Không xác định được kích thước block tối ưu vì dữ liệu có thể ít nhiều khác nhau.

So sánh kết quả SVM song song (có HOG)

- Không có HOG

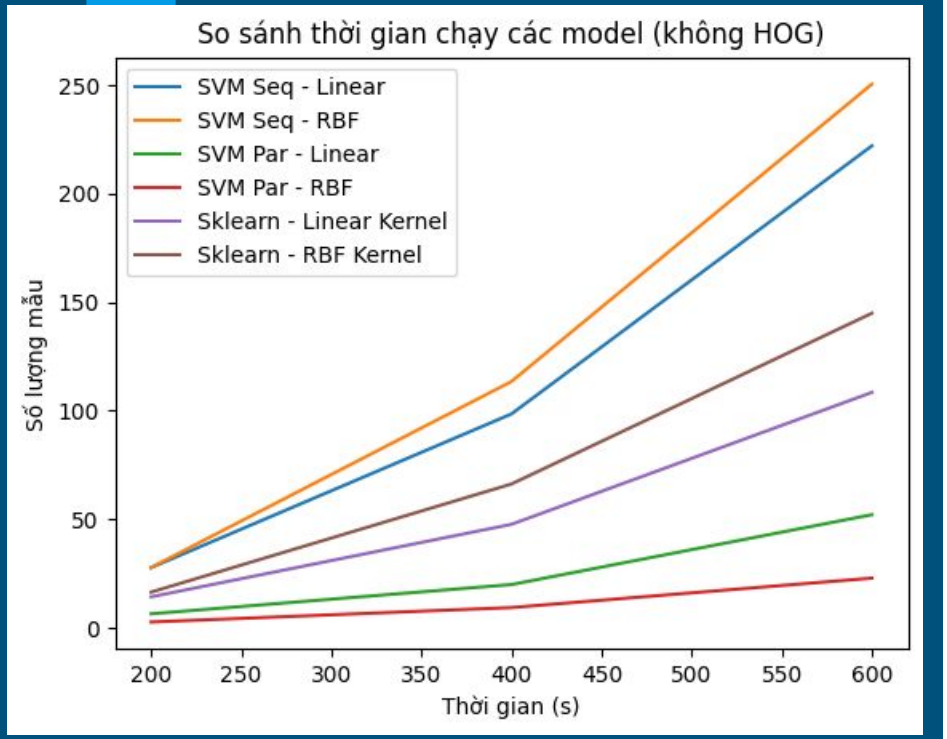
Số lượng mẫu	SVM Seq Linear Acc	SVM Seq RBF Acc	SVM Par Linear Acc	SVM Par RBF Acc	Sklearn Linear+Shrink Acc	Sklearn RBF+Shrink Acc
200	0.616667	0.616667	0.616667	0.616667	0.616667	0.616667
400	0.608333	0.608333	0.608333	0.608333	0.608333	0.608333
600	0.577778	0.655556	0.577778	0.655556	0.577778	0.655556

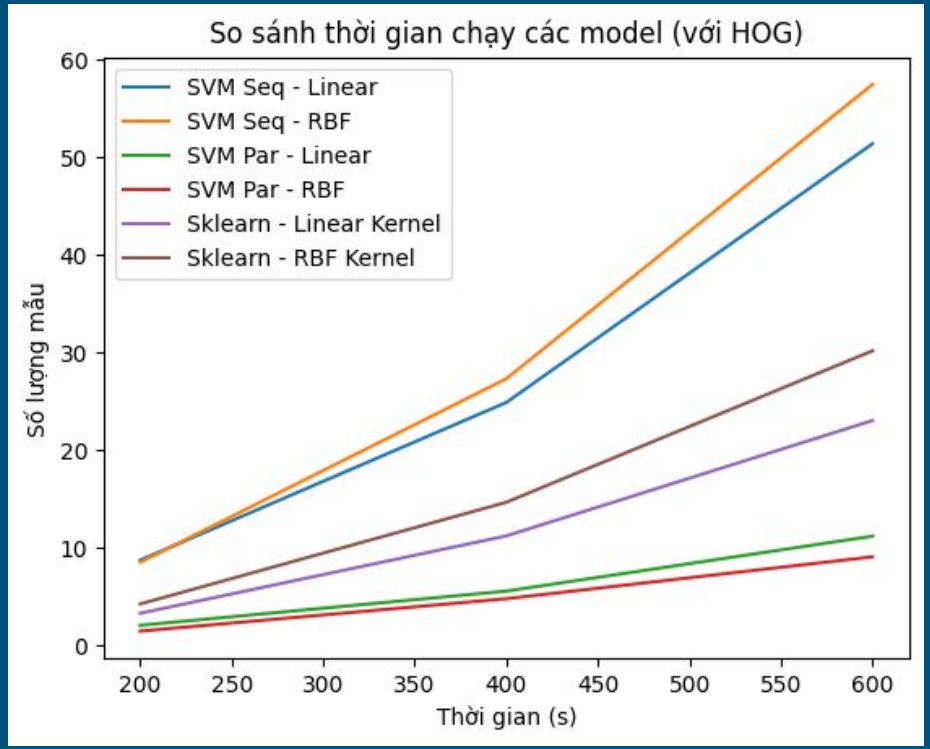
- Có HOG

Số lượng mẫu	SVM Seq Linear + hog Acc	SVM Seq RBF + hog Acc	SVM Par Linear + hog Acc	SVM Par RBF + hog Acc	Sklearn Linear+Shrink + hog Acc	Sklearn RBF+Shrink + hog Acc
200	0.583333	0.583333	0.583333	0.583333	0.583333	0.583333
400	0.633333	0.658333	0.633333	0.658333	0.633333	0.658333
600	0.672222	0.672222	0.672222	0.672222	0.672222	0.672222

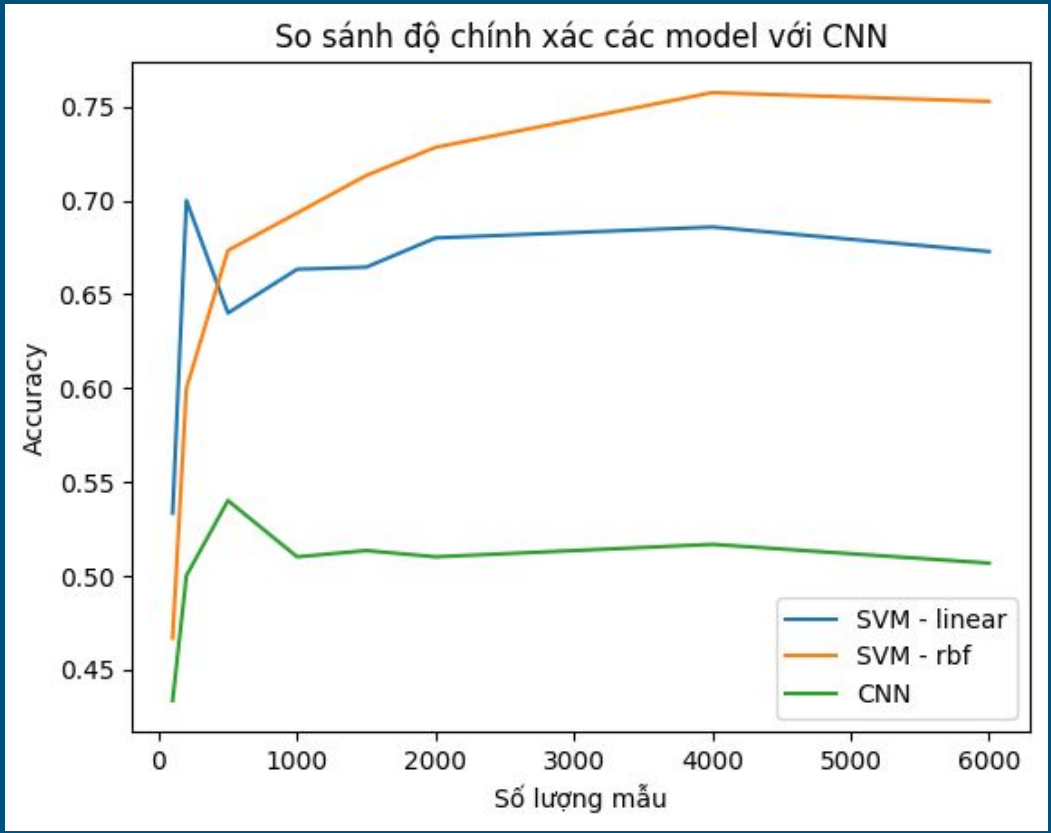
Ta thấy bản song song cho accuracy chính xác trong tất cả trường hợp.

So sánh kết quả SVM song song



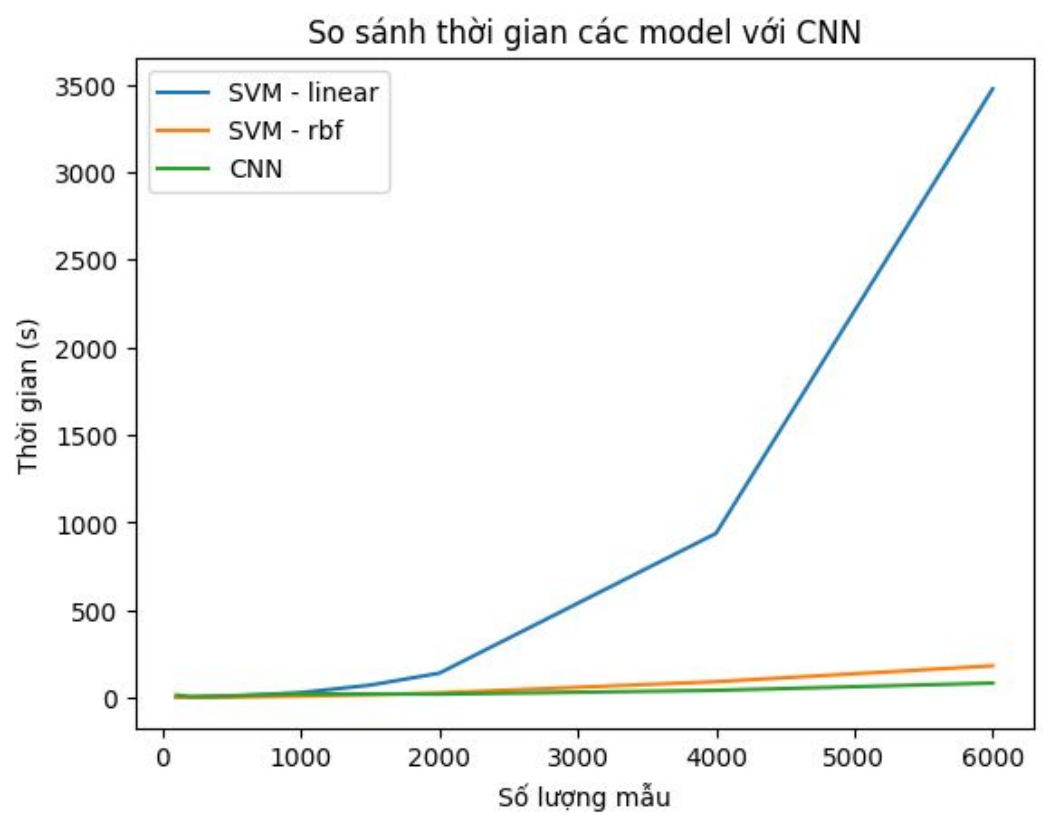


So sánh kết quả SVM+HOG với CNN



- Cả 2 kernel linear và RBF đều tốt hơn nhiều so với CNN.
- CNN cho độ chính xác trung bình khoảng 0.52. Là độ chính xác rất thấp khi dữ liệu test là cân bằng.
- Kernel RBF có độ chính xác tăng khi số lượng dữ liệu tăng

So sánh kết quả SVM+HOG với CNN



- Khi dữ liệu tăng thì kernel linear khó hội tụ hơn nên thời gian chạy lâu hơn.
- CNN của tensorflow khi sử dụng GPU có tốc độ nhỉnh hơn SVM 1 chút ở khoảng > 2000 mẫu.

Tự đánh giá:

- Những gì đã làm được:
  - Xây dựng pipeline hoàn chỉnh gồm 2 model HOG và SVM cho bài toán nhận diện ảnh (Mức 125%)
  - Có các phiên bản song song và cải tiến cả về tốc độ lẫn độ chính xác cho cả 2 thuật toán. Model song song có tốc độ cao hơn nhiều so với thư viện (Mức 125%)
  - Model SVM có accuracy giống 100% so với thư viện sklearn
  - Model HOG và SVM có kết quả cao hơn CNN khi lượng dữ liệu không quá nhiều như đã trình bày ở Seminar 1
  - Thuật toán HOG dù so sánh riêng với thư viện có sự sai khác. Nhưng lại cho kết quả tốt hơn khi áp dụng thực tế vào nhận diện ảnh
- Những gì chưa làm được:
  - Chưa nghĩ ra cách tiết kiệm VRAM khi song song SVM
  - Tối ưu HOG bằng shared memory chưa đạt tốc độ như mong muốn
  - Trễ deadline 2 tuần so với dự kiến
- Tổng kết: nhóm đạt mục tiêu mức 100%