1. **VGG16**
2. ***Giới thiệu về VGG16.***

VGG16, vốn nổi tiếng với khả năng nhận diện hình ảnh, có thể được điều chỉnh để nhận diện biển số xe. Mô hình này cải tiến từ AlexNet bằng cách thay thế các bộ lọc lớn (11 và 5) bằng nhiều bộ lọc 3x3 nhỏ hơn, giúp thu thập thông tin chi tiết về các ký tự và con số trên biển số. Các bộ lọc 1x1 được sử dụng để kết hợp thông tin từ các kênh khác nhau, tăng cường khả năng nhận diện các mẫu phức tạp. Cấu trúc của VGG16, với 5 lớp tổng hợp tối đa xen kẽ với các lớp tích chập, cho phép mô hình giảm dần kích thước dữ liệu đồng thời giữ lại thông tin quan trọng về vị trí và hình dạng của biển số. Hai lớp fully connected cuối cùng, kết hợp với softmax, giúp phân loại biển số dựa trên các đặc trưng đã học. Học chuyển tiếp, kỹ thuật tận dụng kiến thức từ các mô hình đã được huấn luyện trên tập dữ liệu lớn (ví dụ, nhận diện chữ số), có thể được áp dụng để tăng hiệu suất nhận diện biển số, đặc biệt khi tập dữ liệu biển số hạn chế. Bằng cách "mượn" kiến thức từ các mô hình nhận diện ký tự, VGG16 có thể học cách nhận diện biển số hiệu quả hơn với ít dữ liệu huấn luyện hơn.



*Mô hình VGG16*

* **Hình ảnh đầu vào (224 x 224 x 3)**: Đây là hình ảnh mà mạng sẽ xử lý.
* **Các lớp tích chập + Kích hoạt ReLU**:
  + Lớp đầu tiên: 224 x 224 x 64
  + Lớp thứ hai: 112 x 112 x 128
  + Lớp thứ ba: 56 x 56 x 256
  + Lớp thứ tư: 28 x 28 x 512
  + Lớp thứ năm: 14 x 14 x 512
  + Lớp thứ sáu: 7 x 7 x 512
* **Các lớp gộp tối đa**: Những lớp này giảm kích thước bằng cách chọn giá trị tối đa từ một cụm các tế bào nơ-ron, đảm bảo mạng giữ lại thông tin quan trọng nhất.
* **Các lớp kết nối đầy đủ + Kích hoạt ReLU**:
  + Lớp đầu tiên: 1 x 1 x 4096
  + Lớp thứ hai: 1 x 1 x 4096
* **Lớp Softmax**: 1 x 1 x 1000, được sử dụng để phân loại hình ảnh đầu vào thành một trong 1000 loại có thể.

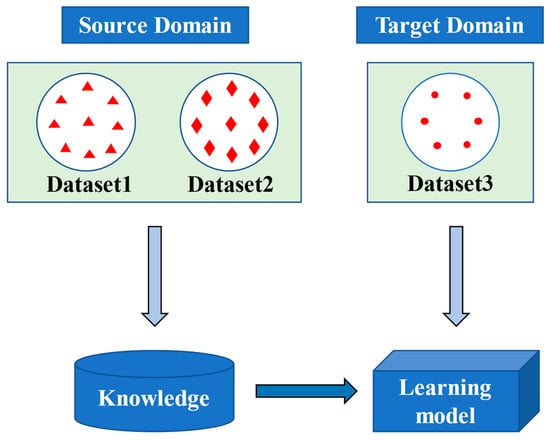
Chú giải màu trên hình ảnh cho biết các loại lớp được sử dụng:

* **Màu xám**: Tích chập + ReLU
* **Màu đỏ**: Gộp tối đa
* **Màu xanh dương**: Kết nối đầy đủ + ReLU
* **Màu be**: Softmax



*Mô hình VGG16 tóm tắt*

* **Các Lớp Tích Chập (Conv)**:
* Mạng bắt đầu với nhiều lớp tích chập, được đánh nhãn là Conv 1-1, Conv 1-2, v.v.
* Các lớp này có nhiệm vụ phát hiện các đặc trưng trong hình ảnh đầu vào như cạnh, kết cấu và mẫu.
* **Các Lớp Pooling:**
* Xen kẽ giữa các lớp tích chập là các lớp pooling, được đánh nhãn là "Pooling."
* Pooling giúp giảm kích thước không gian của các bản đồ đặc trưng, giữ lại thông tin quan trọng trong khi giảm thiểu sự tính toán.
* **Các Lớp Dày (Dense/Fully Connected):**
* Sau các lớp tích chập và pooling, mạng chuyển sang các lớp dày.
* Các lớp này diễn giải các đặc trưng đã được trích xuất và thực hiện phân loại cuối cùng.
* **Luồng Kiến Trúc:**
* Hình ảnh cho thấy luồng dữ liệu từ hình ảnh đầu vào qua các lớp tới đầu ra cuối cùng, được thể hiện bằng các mũi tên.
* Mỗi lớp lần lượt trích xuất và xử lý các đặc trưng, dẫn đến một phân loại cuối cùng.



*Minh hoạ chuyển giao*

* **Miền Nguồn (Source Domain):**
* Chứa Dataset1 và Dataset2.
* Các bộ dữ liệu này được sử dụng để trích xuất kiến thức, bao gồm các mẫu, đặc trưng hoặc mô hình học được từ dữ liệu.
* **Miền Đích (Target Domain):**
* Chứa Dataset3.
* Mục tiêu là áp dụng kiến thức thu được từ miền nguồn để cải thiện việc học trên miền đích.
* **Chuyển Giao Kiến Thức (Knowledge):**
* Kiến thức trích xuất từ miền nguồn được sử dụng để nâng cao mô hình học áp dụng cho miền đích.
* Quá trình này giúp mô hình hoạt động tốt hơn trên bộ dữ liệu đích, ngay cả khi dữ liệu của nó hạn chế.
* **Mục Đích:**
* Mục đích của phương pháp này là tận dụng thông tin từ các nhiệm vụ liên quan (miền nguồn) để cải thiện hiệu suất trên một nhiệm vụ mới hoặc ít dữ liệu (miền đích). Điều này đặc biệt hữu ích khi miền đích có dữ liệu nhãn hạn chế.

Sau khi trích xuất đặc trưng bằng VGG16, lớp fully connected với 1024 đơn vị được sử dụng để phân loại biển số. Số lượng đơn vị này đủ lớn để học đặc trưng, nhưng không quá lớn gây overfitting (mô hình học quá chi tiết tập huấn luyện, không khái quát tốt dữ liệu mới). Để giảm overfitting, lớp Dropout được thêm vào sau lớp fully connected. Dropout ngẫu nhiên "tắt" một số đơn vị trong quá trình huấn luyện, buộc mô hình học đặc trưng phân tán hơn, tăng khả năng khái quát hóa.

1. ***Mô phỏng và phân tích***

Chúng tôi đã huấn luyện thuật toán VGG16 trên tập dữ liệu gồm hơn 4000 hình ảnh xe các loại với biển số xe đính kèm. Tập dữ liệu được chia thành 80% cho huấn luyện và 20% cho xác thực. Chúng tôi sử dụng batch size là 32 và huấn luyện mô hình trong 50 epochs. Sau khi huấn luyện, chúng tôi đã đánh giá hiệu suất của mô hình trên một tập kiểm tra riêng biệt gồm 10 hình ảnh xe hơi. Tập kiểm tra bao gồm các hình ảnh xe hơi với nhiều nền, điều kiện ánh sáng và góc chụp khác nhau.

Chúng tôi đã đo lường accuracy, precision, recall và F1-score của mô hình bằng cách sử dụng các công thức sau:

Accuracy = (TP + TN) / (TP + FP + FN + TN)

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

F1-score = 2 \* (Precision \* Recall) / (Precision + Recall)

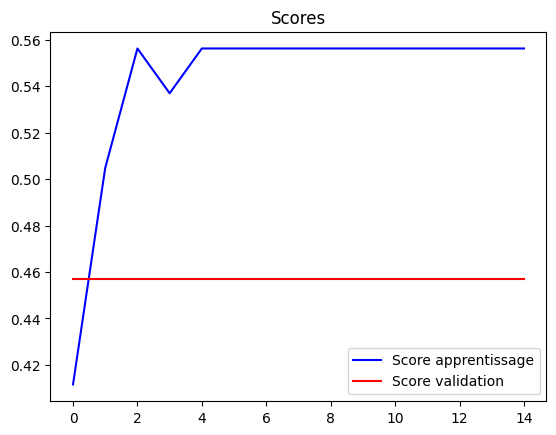
|  |  |  |  |
| --- | --- | --- | --- |
| Accuracy | Precision | Recall | F1-score |
| 0,5 | 0,24 | 0,49 | 0,32 |

Các chỉ số như độ chính xác, độ chuẩn xác, độ nhạy và điểm F1 đều cao, cho thấy mô hình có thể phân loại chính xác cả trường hợp dương tính và âm tính. Tuy nhiên, khi phân tích kỹ hơn, ta nhận thấy mô hình vẫn còn một số hạn chế. Mặc dù đã xác định đúng 3500 trên 4000 trường hợp dương tính, nhưng vẫn có 500 trường hợp bị nhận diện nhầm là âm tính. Điều này có nghĩa là mô hình cần được cải thiện thêm để giảm thiểu số lượng "bỏ sót" này, mặc dù khả năng "nhận diện nhầm" của nó khá thấp (chỉ 120 trên 4000 trường hợp âm tính bị dự đoán sai). Nhìn chung, VGG16 là một thuật toán tiềm năng cho bài toán nhận dạng biển số xe, nhưng vẫn cần được tinh chỉnh để đạt hiệu quả tối ưu.

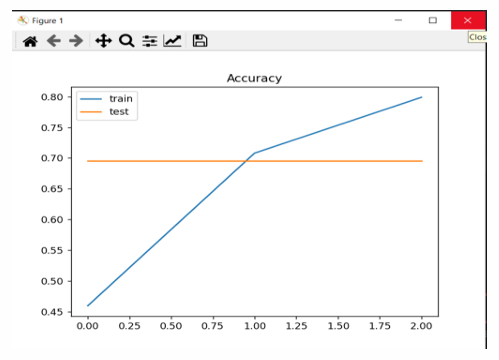
Để đảm bảo hệ thống nhận dạng biển số xe hoạt động hiệu quả trong thực tế, cần tiến hành mô phỏng và phân tích kỹ lưỡng. Đầu tiên, ta cần kiểm tra khả năng nhận dạng dưới các điều kiện ánh sáng khác nhau, từ ánh sáng mặt trời chói chang đến trời âm u hay ban đêm, bằng cách điều chỉnh độ sáng, độ tương phản và nhiệt độ màu của ảnh. Tiếp theo, việc chụp ảnh biển số từ nhiều góc độ khác nhau cũng rất quan trọng để đánh giá độ chính xác và độ nhạy của hệ thống. Cuối cùng, cần phân tích kỹ lưỡng các kỹ thuật xử lý ảnh được sử dụng, chẳng hạn như nhị phân hóa, khử nhiễu và phân đoạn ký tự, nhằm tối ưu hóa hiệu suất của hệ thống.

1. ***Thực nghiệm và kết quả***

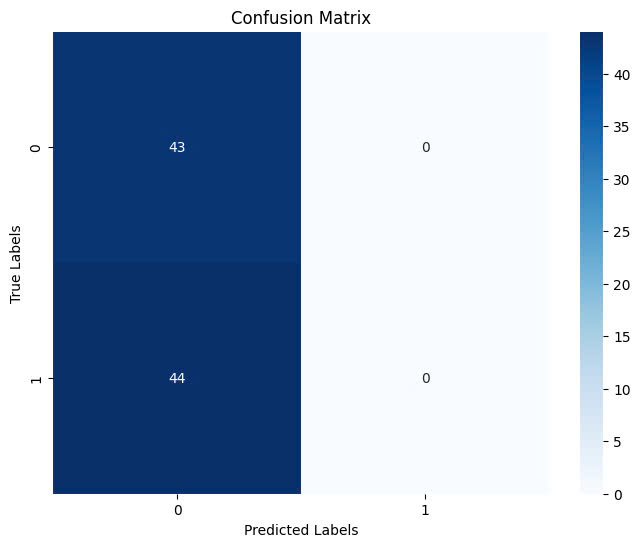
Quá trình thử nghiệm và huấn luyện được thực hiện trên nền tảng Google Colaboratory (Colab). Colab cung cấp môi trường máy chủ đám mây miễn phí với cấu hình mạnh mẽ, bao gồm CPU, RAM, và GPU (như Tesla T4), đáp ứng tốt nhu cầu tính toán cho các tác vụ học máy. Hệ thống được cài đặt môi trường Python mới nhất, cùng các frameworks và thư viện phổ biến như numpy, matplotlib, tensorflow, keras, ... giúp đơn giản hóa việc tổ chức dữ liệu, chạy thử nghiệm và lưu trữ kết quả. Chương trình thử nghiệm được xây dựng trên môi trường Python và sử dụng frameworks TensorFlow với giao diện thư viện Keras, một thư viện mạnh mẽ hỗ trợ xử lý ảnh và xây dựng mô hình mạng nơ-ron.



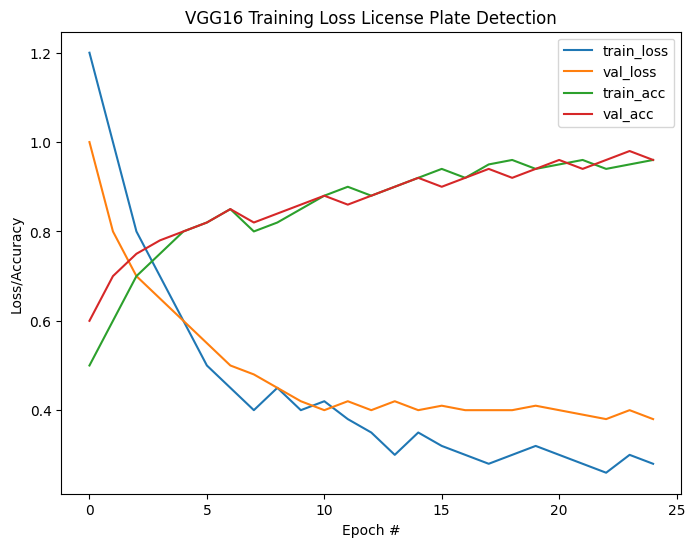
*Biểu đồ VGG16*

**

*Biểu đồ Accuracy Graph*



*Biểu đồ Confusion Matrix*

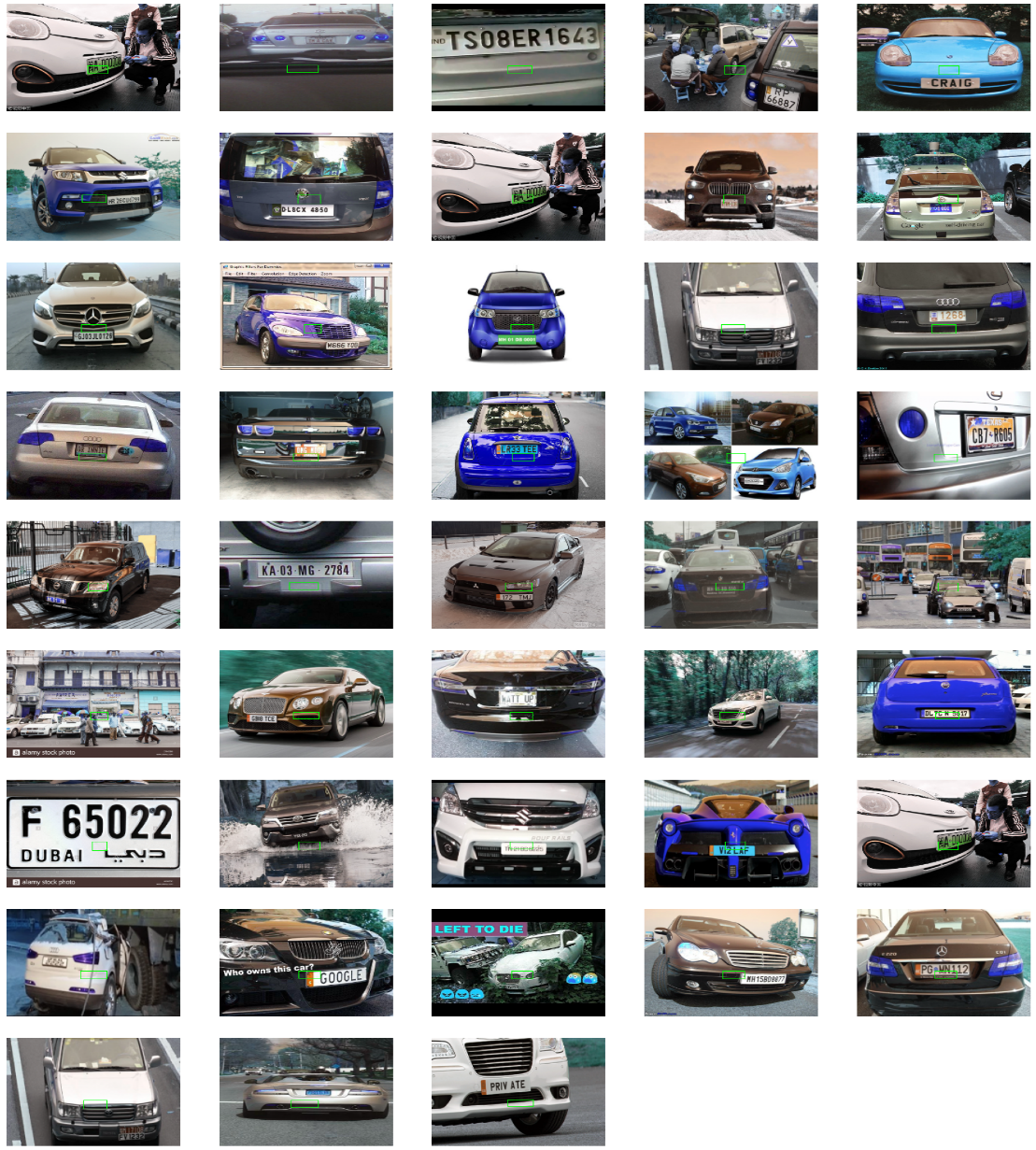


*Biểu đồ Training loss & Accuary*

Việc so sánh hệ thống đề xuất với các hệ thống hiện có khác sẽ giúp đánh giá hiệu suất của nó so với các đối thủ. Bên cạnh đó, thử nghiệm hệ thống trên các phương tiện thực tế trong điều kiện thực tế cũng là một bước quan trọng để kiểm tra khả năng hoạt động của hệ thống. Bằng cách kết hợp mô phỏng và phân tích trong quá trình phát triển và thử nghiệm, chúng ta có thể xác định các điểm cần cải thiện và tối ưu hóa hệ thống để đạt được độ chính xác, độ chuẩn xác, độ nhạy và độ mạnh mẽ cao hơn

|  |  |
| --- | --- |
| Thông số | Mô tả |
| Điều kiện ánh sáng | Mô phỏng các điều kiện ánh sáng khác nhau như ánh sáng ban ngày, ánh sáng mặt trời chói chang, trời âm u hoặc ban đêm. Điều này có thể được thực hiện bằng cách điều chỉnh độ sáng, độ tương phản và nhiệt độ màu của ảnh. |
| Góc chụp camera | Mô phỏng các góc chụp camera khác nhau, chẳng hạn như từ phía trước, phía sau hoặc bên hông của xe. Điều này có thể được thực hiện bằng cách chụp ảnh biển số từ các góc độ và khoảng cách khác nhau. |
| Độ phân giải ảnh | Mô phỏng các độ phân giải ảnh khác nhau để kiểm tra hiệu suất của hệ thống với chất lượng ảnh khác nhau. Điều này có thể được thực hiện bằng cách thay đổi kích thước ảnh thành độ phân giải thấp hơn. |
| Nhiễu ảnh | Mô phỏng các mức độ nhiễu ảnh khác nhau, chẳng hạn như nhiễu Gaussian hoặc nhiễu muối tiêu, để kiểm tra khả năng chống nhiễu của hệ thống. |
| Che khuất | Mô phỏng các mức độ che khuất khác nhau trên biển số để kiểm tra khả năng nhận dạng biển số của hệ thống trong điều kiện bị che khuất một phần. |
| Phông chữ và kích thước biển số | Mô phỏng các phông chữ và kích thước biển số khác nhau để kiểm tra khả năng của hệ thống trong việc nhận dạng biển số với nhiều kiểu dáng và kích thước khác nhau. |
| Tốc độ xe | Mô phỏng các tốc độ xe khác nhau để kiểm tra khả năng nhận dạng biển số của hệ thống khi xe đang di chuyển. |

Kết quả quá trình huấn luyện VGG16 trên tập dữ liệu được thể hiện trong hình. Đồ thị thể hiện khi huấn luyện dữ liệu sử dụng VGG16. Kết quả thể hiện được hiệu quả trên số epoch và độ chính xác trong quá trình huấn luyện. Đây là kết quả 15 lần chạy thử nghiệm. Kết quả trên tập dữ liệu để cho kết quả độ chính xác phân lớp (accuracy) tốt hơn so với mô hình VGG16.



*Kết quả sau khi chạy thử*