**گزارش پروژه تشخیص اشیاء با استفاده از CNN و DenseNet بر روی داده‌های CIFAR-10**

**مقدمه**

در این پروژه، با هدف تشخیص اشیاء در تصاویر، به پیاده‌سازی و مقایسه دو مدل یادگیری عمیق پرکاربرد، یعنی CNN و DenseNet، بر روی مجموعه داده تصویری CIFAR-10 پرداخته شد. جزئیات دقیق پیاده‌سازی هر یک از این مدل‌ها، از جمله ساختار شبکه، فرایند آموزش، و نتایج حاصل، در فایل‌های Jupyter Notebook مربوطه ارائه شده است.

**مجموعه داده CIFAR-10**

مجموعه داده CIFAR-10 شامل 60000 تصویر رنگی با ابعاد 32 در 32 پیکسل است که به ده کلاس مختلف از اشیاء تقسیم‌بندی شده‌اند. این مجموعه داده به دلیل حجم مناسب و تنوع اشیاء، به طور گسترده برای ارزیابی مدل‌های یادگیری عمیق در حوزه بینایی ماشین مورد استفاده قرار می‌گیرد.

**معماری‌های مورد استفاده**

* **CNN:** یک معماری پایه کانولوشنال عمیق با تعدادی لایه کانولوشن، لایه‌های جمع‌کننده و لایه‌های تمام‌متصل طراحی و پیاده‌سازی شد. جزئیات دقیق این معماری، از جمله تعداد لایه‌ها، تعداد فیلترها و اندازه هسته‌های کانولوشن، در فایل Jupyter Notebook مربوطه قابل مشاهده است.
* **DenseNet:** از یک مدل DenseNet پیش‌آموزش‌یافته به عنوان نقطه شروع استفاده شد و سپس با توجه به ویژگی‌های مجموعه داده CIFAR-10، تنظیماتی بر روی آن انجام شد. ساختار دقیق این مدل، از جمله تعداد بلوک‌ها، تعداد لایه‌ها در هر بلوک و نرخ رشد، در فایل Jupyter Notebook مربوطه آورده شده است.

**مراحل انجام پروژه**

1. **آماده‌سازی داده‌ها:**
   * بارگذاری و پیش‌پردازش داده‌های CIFAR-10 شامل نرمال‌سازی، افزایش داده‌ها (Augmentation) و تقسیم‌بندی داده‌ها به مجموعه‌های آموزش، اعتبارسنجی و تست.
2. **طراحی و پیاده‌سازی مدل:**
   * طراحی و پیاده‌سازی معماری‌های CNN و DenseNet در محیط TensorFlow/Keras.
   * تعریف تابع هزینه (Loss function) مناسب (مثلاً Cross-Entropy) و انتخاب بهینه‌ساز (مثلاً Adam).
3. **آموزش مدل:**
   * آموزش مدل‌ها بر روی مجموعه داده آموزش با استفاده از الگوریتم‌های بهینه‌سازی مناسب.
   * ارزیابی عملکرد مدل در هر دوره آموزش بر روی مجموعه اعتبارسنجی.
4. **ارزیابی مدل:**
   * ارزیابی عملکرد نهایی مدل‌های آموزش دیده بر روی مجموعه تست با استفاده از معیارهای ارزیابی مانند دقت (Accuracy)، دقت هر کلاس (Precision)، فراخوانی (Recall) و F1-score.
5. **مقایسه نتایج:**
   * مقایسه عملکرد دو مدل CNN و DenseNet و تحلیل دلایل تفاوت در عملکرد آن‌ها.

**نتایج**

پس از آموزش و ارزیابی هر دو مدل بر روی مجموعه داده تست، نتایج زیر حاصل شد:

* **مدل DenseNet:**
  + **تابع هزینه (Loss):** 1.4368
  + **دقت (Accuracy):** 0.4903
* **مدل CNN:**
  + **تابع هزینه (Loss):** 0.7912
  + **دقت (Accuracy):** 0.7293

همانطور که مشاهده می‌شود، مدل CNN در این آزمایش عملکرد بهتری نسبت به مدل DenseNet داشته است. یکی از دلایل احتمالی این تفاوت، پیچیدگی بیشتر ساختار DenseNet و نیاز آن به داده‌های آموزشی بیشتر برای آموزش مناسب است. همچنین، تنظیم دقیق پارامترهای مدل DenseNet (مانند نرخ یادگیری، تعداد لایه‌ها و ...) می‌تواند تأثیر قابل توجهی بر عملکرد آن داشته باشد.

**تحلیل عمیق‌تر**

برای درک بهتر دلایل این تفاوت در عملکرد، می‌توان به مقایسه معماری‌های دو مدل و پارامترهای آموزش آن‌ها پرداخت. به عنوان مثال، می‌توان تعداد لایه‌ها، تعداد فیلترها، نوع تابع فعال‌سازی و بهینه‌ساز مورد استفاده در هر دو مدل را مقایسه کرد. همچنین، می‌توان به بررسی نمودارهای یادگیری (learning curves) دو مدل پرداخت تا روند کاهش تابع هزینه و افزایش دقت در طول آموزش را مشاهده کرد.

**پیشنهادات برای کارهای آینده**

* **افزایش حجم داده‌های آموزشی:** افزایش حجم داده‌های آموزشی می‌تواند به بهبود عملکرد هر دو مدل، به ویژه مدل DenseNet، کمک کند.
* **استفاده از تکنیک‌های افزایش داده (Data Augmentation):** اعمال تکنیک‌هایی مانند چرخش، برش، تغییر اندازه و تغییر روشنایی بر روی تصاویر می‌تواند به افزایش تنوع داده‌های آموزشی و بهبود عملکرد مدل‌ها کمک کند.
* **بهینه‌سازی معماری مدل‌ها:** می‌توان با تغییر در معماری مدل‌ها، مانند استفاده از بلوک‌های کانولوشن مختلف یا تغییر تعداد لایه‌ها، به عملکرد بهتری دست یافت.
* **استفاده از تکنیک‌های تنظیم فوق‌پارامتر (Hyperparameter Tuning):** با استفاده از تکنیک‌هایی مانند Grid Search یا Randomized Search می‌توان پارامترهای بهینه مدل را پیدا کرد.
* **بررسی مدل‌های دیگر:** می‌توان مدل‌های تشخیص اشیاء دیگری مانند Faster R-CNN، YOLO و Mask R-CNN را بررسی کرده و نتایج آن‌ها را با مدل‌های مورد استفاده در این پروژه مقایسه کرد.

**نتیجه‌گیری**

در این پروژه، به مقایسه عملکرد دو مدل CNN و DenseNet در تشخیص اشیاء پرداخته شد. نتایج حاصل نشان داد که در این حالت خاص، مدل CNN عملکرد بهتری داشته است. با این حال، انتخاب مدل مناسب به عوامل مختلفی بستگی دارد و ممکن است در پروژه‌های دیگر، نتایج متفاوتی حاصل شود.

**توجه:** برای مشاهده جزئیات بیشتر در مورد پیاده‌سازی مدل‌ها، نتایج آزمایش‌ها و کدهای مربوطه، لطفا به فایل‌های Jupyter Notebook مراجعه نمایید.