

Genre (Multi Class) Classification with Movie Images using Image Embeddings with PreTrained Models for BLM 5103 Computer Vision Project

BLM5103 Bilgisayarlı Görü Dersi Dönem Sonu Projesi

Metin Uslu & Anıl Özcan & Hamza Cantürk

Özet: Bu çalışmada amacımız film posterlerini kullanarak film kategorisini tahmin edebilmektir. Multi Class Classification, Transfer Learning, Pre Trained modelleri kullanarak posterlerin embedding'lerinin elde edilmesi ve bu embeddinglerin tahmin modellerinde kullanılması amaçlanmıştır.

Giriş: Üzerinde çalıştığımız problem film posterlerine bakarak bu filmlerin kategorilerini tahmin edebilmektir. Kategoriler suç, drama, aksiyon vb. dallardır.

İlgili çalışmalar: Proje için teklif verme süresince konu araştırırken ve veri toplamak için araştırma yapılırken bu veri seti üzerinde bizim çalıştığımız konular hakkında kapsamlı bir çalışmaya rastlamadık.

Veri Hakkında

Veri seti olarak Kaggle sayfasından <https://www.kaggle.com/datasets/neha1703/movie-genre-from-its-poster?select=SampleMoviePosters> kullanılmıştır. Film izleyicileri için film posterleri, insanların filmin içeriği ve türü hakkında ipuçları almak için kullandıkları ilk izlenimlerden biridir. İnsanlar türü (korku, komedi, animasyon vb.) hızlı bir şekilde belirlemek için renk, oyuncuların yüzlerindeki ifadeler vb. gibi ipuçlarını kavrayabilirler. Bir görüntünün renk tonu, doygunluk, parlaklık vb. renk özelliklerinin insan duygularını etkilediği gösterilmiştir. Belirli bir durum insanlarda bu duyguları uyandırır. Eğer insanlar bir filmin türünü posterine tek bir bakışla tahmin edebiliyorsa, o zaman posterlerin renk özelliklerinin, yerel doku tabanlı özelliklerinin ve yapısal ipuçlarının, derin öğrenme algoritmalarında filmin türünü tahmin etmek için kullanılabilecek bazı özelliklere sahip olduğunu varsayabiliriz.

Film afişleri IMDB sitesinden alınmıştır. Toplanan veri kümesinde IMDB Kimliği, IMDB Bağlantısı, Başlık, IMDB Puanı, Tür ve film posterlerini indirme bağlantısı bulunur. Her Film posterini en az bir türe ait olabilir ve kendisine en fazla 3 tür etiketi atanabilir. Veri setine source code altında "data/input/ MovieGenre.csv" altından ulaşabilirsiniz. Yalnız bu MovieGenre.csv veri seti 40108x6 satırdan oluşmaktadır.

Metotlar

İlk aşamada, Inception ve ResNet modelleri kullanılarak film afişi veri kümesinden gömülü vektörler oluşturuldu. Bu vektörler, posterlerin karmaşık görsel özelliklerini ve temsillerini kapsayarak daha sonra analiz ve model geliştirme sürecinde zengin gömülü vektörler sağlama amacını taşıdı. Transfer öğrenimi kullanılarak VGG19 modeli, önceden elde edilen gömülü vektörlerle tahmin yapmak için kullanıldı. Gömülü vektörleri VGG19 mimarisine girdi olarak alarak, modelin öğrenilen temsillere dayanarak film türlerini tahmin etme becerisini geliştirmeyi hedefledi. Bu adım, VGG19 modelinin tür sınıflandırması konusundaki tahmin yeteneklerini iyileştirmek için öğrenilen gömülü vektörlerden faydalandı. Çıkarılan gömülü vektörler, derin öğrenme modellerinin performansını artırmak için kullanıldı. Bu vektörler, görsel özelliklerin zenginliğini taşıyarak derin öğrenme modellerinin yapısına entegre edildi ve film afişi görüntülerindeki karmaşık desenleri yakalama kabiliyetini artırmayı amaçladı. Bu süreç, modellerin sınıflandırma doğruluğunu artırmak ve geliştirmek için gömülü vektörleri kullanma odaklıydı. Ayrıca, ResNet50 mimarisi üzerinde yapılan transfer öğrenimi çalışması, daha önce elde edilen gömülü vektörlerin kullanılmadan yapılmasını içeriyordu. Bu çalışma, ResNet50 modelinin film afiş veri kümesinde bulunan özgül detaylara daha iyi uyum sağlamasını hedefledi ve bu amaçla modelin eğitimi yapıldı. Ayrıca gömülü vektörler kullanılarak elde edilen özellik vektörleri ve sınıfları ile bir Multi-Class classification modeli kurulup tahmin başarısı da ölçülmüştür.

Deneyler

RestNet50

Projenin birincil amacı, derin öğrenme tekniklerini kullanarak, özellikle ResNet50, VGG19 gibi mimarilerden yararlanarak verimli bir görüntü sınıflandırma sistemi geliştirmektir. Amaç, görüntüleri belirli bir veri kümesi içinde önceden tanımlanmış kategorilere veya sınıflara doğru bir şekilde sınıflandırmaktır. Bu sınıflandırma sistemi, görsel verilerin tanımlanması ve kategorize edilmesinde bilgisayarla görmenin pratik uygulamasını sergilemeyi amaçlamıştır. Raporun bu kısmında ResNet50 üzerine yapılan denemeler ve sonuçları aktarılacaktır.

Bu projede kullanılan veri kümesi, her biri farklı sınıflara ait çeşitli kategorilerdeki görüntülerden oluşuyordu. Veri kümesinin özellikleri şunları içeriyordu:

Boyut: Veri kümesi, toplamda 1000 görüntüden oluşan ortalama sayıda görüntüden oluşuyordu.

Format: Görüntüler PNG gibi standart bir formatta saklanarak görüntü işleme kütüphaneleri ile uyumluluk sağlanmıştır.

Sınıflar: Veri kümesi, çeşitli sınıflandırma görevlerini mümkün kılmak için film kategorileri arasında değişen birden fazla sınıfı veya kategoriye kapsıyordu.

Çözünürlük: Görüntüler, [genişlik x yükseklik x renk kanalları] anlamına gelen [162x268x3] çözünürlüğe sahiptir.

ResNet50 ve Transfer Öğrenimini Seçme Motivasyonu:

ResNet50, özellikle görüntü sınıflandırma görevlerinde kullanılan 50 katmanlı bir evrişimli sinir ağı olarak öne çıkıyor. Bu ağın derinliği ve mimarisi, görüntü işleme alanında etkili sonuçlar elde etmek için güçlü bir araç sunuyor. Özellikle, derin ağın içerdiği bağlantılar, eğitim sırasında karşılaşılan kaybolan gradyan gibi sorunları azaltarak daha sağlam bir eğitim süreci sunuyor. ResNet50'nin tercih edilmesindeki nedenlerden biri de transfer öğrenimi imkanlarının bulunmasıdır. Büyük ölçekli veri kümeleri, örneğin ImageNet gibi geniş ve çeşitli veri setlerinde eğitilen bu ağın önceden belirlenmiş ağırlıkları, daha küçük ve özelleşmiş veri setlerinde genelleştirilmiş özelliklerin öğrenilmesine yardımcı olur. Transfer öğrenimi, eğitim sürecini hızlandırırken, modelin hedeflenen alandaki performansını artırmak için önceden edinilen bilgilerden yararlanır. Bu prensipleri temel alan bir proje, ResNet50 modelini seçerek ve transfer öğrenimini kullanarak önceden eğitilmiş derin öğrenme modellerinin gücünden yararlanmayı ve bu modelleri belirli, daha sınırlı veri kümelerine uyarlamayı hedefler. Bu yaklaşım, sınırlı veri ve hesaplama kaynaklarına rağmen güçlü bir görüntü sınıflandırma performansı elde etme potansiyeline sahiptir. Yani, daha az veri veya işlem gücü ile bile etkili sonuçlar alınabilir. Sonuç olarak, bu strateji, önceden eğitilmiş modellerin avantajlarını kullanarak özelleştirilmiş bir öğrenme süreci yürütür. Bu da, verimli, doğru ve özelleştirilmiş bir görüntü sınıflandırma sistemi oluşturma amacına yönelik bir yaklaşımı ifade eder. Bu sayede, belirli bir uygulama alanına odaklanan ve bu alandaki verileri daha etkili bir şekilde işleyebilen bir model elde etmek mümkün olabilir.

Veri Ön İşleme

Image embeddinglerin elde edilmesi:

Bu kısımda Python ve IDE olarak Jupyter Notebook kullanarak Film Görseller ile MovieGenre veri seti üzerinde bazı preprocess işlemleri gerçekleştirdik. Aşağıda tüm yapılanlara DataPreProcess.ipynb üzerinden ulaşabilirsiniz. Bunlardan bazıları;

1. MovieGenre veri seti üzerinde;
 - a. Tip dönüşümleri (Type Casting) uygun hale getirilerek yapılarak hem memory'de daha az yer tüketmesini hem de tabloların olası birleştirilmesi (join) süreçlerinde aynı tipte olması sağladık.
 - b. Duplicate olan satırların (593 Adet) hariç tutulması (exclude) sağladık. 39515 x 6 satırlık bir veri seti haline dönüştü.
2. "data/input/images" altında yer alan Film Görselleri ile bir veri seti oluşturduk. Bu veri seti 2 "ImgName", "ImgId" kolonlarından oluşmaktadır. ImgName dosya isimleri (filename+extension), ImgId ise dosya isimlerinden id bilgilerinin yer aldığı bilgilerini tutmaktadır.
3. İki veri setinin "Img" kolonu üzerinden birleştirilerek Film Görselleri ile Film Detay bilgileri tek bir veri seti üzerinde toplanmıştır. Bu join işlemi sonucunda 997 x 8 bir veri seti elde edilmiştir.
4. Genre kolonu bir film için bir ya da birden ya da kategori bilgisi içerebilmektedir. Örneğin "Richard III" filmi için "Drama" iken "Now and Forever (1934)" filmi için Drama|Romance olarak 2 kategorisi bulunmaktadır. Bu çalışmada MultiClass

Classification yapacağımız için bir film için bir tane kategorinin bulunması gerekiyor. Genre kolonu | sembolü ile split edilerek Genre1, Genre2, ... , GenreN kolonları oluşturulmuştur.

5. Genre kolonunda yer alan "Short" film türü filmin kısa/uzun bilgisi içerdiği ve film görsellerinde bunun yansıtılmış olmasını aramanın/beklemenin yanlış olacağını düşünerek "Short" bilgisine sahip filmlerde aynı filme ait 2. Genre türü ile replace edilmiştir.
6. Short film türünü replace ettikten sonra veri setimizde 18 tane farklı film türü bulunmaktadır. Bu çalışmada bazı film türleri için veri azlığından kaynaklı modelin performansını olumsuz yönde etkilemesi söz konusu olduğu için veri setinin %81.34 içeren ilk 4 film türü "Dram, Comedy, Crime, Adventure" film türleri seçilerek filtrelenmiştir. Film türü istatistik bilgileri aşağıda paylaşılmıştır.

[21]:

	Genre1	Cnt	CntRatio	CumSum	CumRatio
7	Drama	357	35.81	357	35.81
4	Comedy	317	31.80	674	67.60
5	Crime	84	8.43	758	76.03
1	Adventure	53	5.32	811	81.34
0	Action	38	3.81	849	85.16
2	Animation	23	2.31	872	87.46
3	Biography	20	2.01	892	89.47
6	Documentary	17	1.71	909	91.17
15	Romance	16	1.60	925	92.78
9	Fantasy	14	1.40	939	94.18
11	Horror	13	1.30	952	95.49
14	Mystery	12	1.20	964	96.69
17	Western	11	1.10	975	97.79
13	Musical	8	0.80	983	98.60
8	Family	3	0.30	986	98.90
16	War	3	0.30	989	99.20
10	History	2	0.20	991	99.40
12	Music	2	0.20	993	99.60

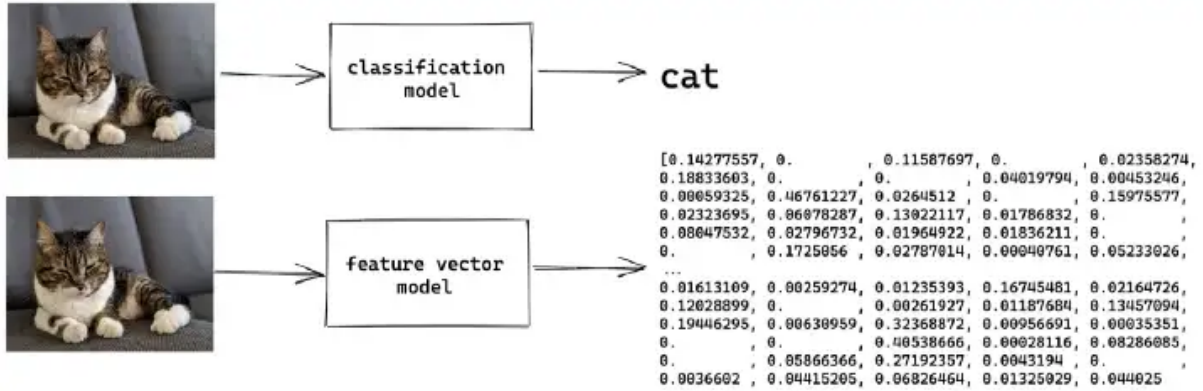
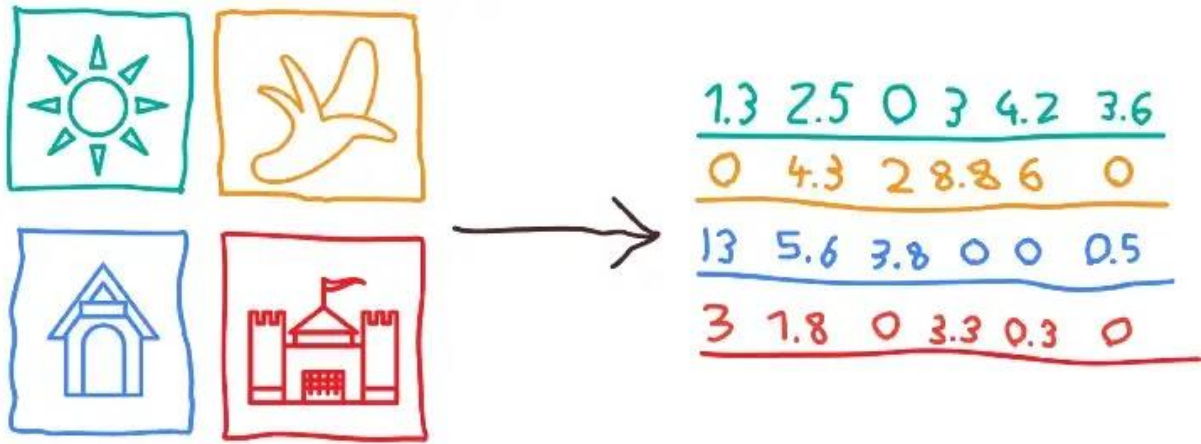
Bu adımdan sonra veri setimiz şekillenmeye başladı. Artık elimizde Görseller ve bu görsellerin film türlerini tuttuğumuz 811 x 9 olmak üzere veri setimiz bulunmaktadır. Diskte daha az yer

tutması için csv.gz olarak “data/preprocess/MasterData.csv.gz” dosyası üzerinden ulaşabilirsiniz.

Image Embedding (Image Feature Vector)

Görüntü gömme/yerleştirmeler, görüntülerin anlamsal anlamlarını ve görsel özelliklerini yakalayan sayısal temsilleridir. Görsel arama, görsel sınıflandırma, görsel alma ve görsel benzerliği gibi birçok uygulama için faydalıdır.

Görüntü gömme/yerleştirme, yüksek boyutlu bir uzayda bir görüntüyü temsil eden bir sayı vektörüdür. Örneğin, bir kedinin görüntüsü [0.12, -0.34, ..., 0.05] gibi 384 sayıdan oluşan bir vektör olarak gömülebilir. Vektördeki her sayı, görüntünün renk, şekil, doku vb. gibi bir özelliğine veya niteliğine karşılık gelir. Vektör, görüntünün özünü yakalar ve onu matematiksel işlemler kullanarak diğer görüntülerle karşılaştırmamıza olanak tanır.



Görüntü yerleştirmelere aynı zamanda görüntü özellikleri (image features) veya görüntü tanımlayıcıları (image descriptors) da denir. Bunlar, onlardan anlamlı bilgilerin nasıl çıkarılacağını öğrenmek için büyük görüntü veri kümeleri üzerinde eğitilmiş görüntü modellerinden türetilir. Evrişimli sinir ağları (CNN'ler), otomatik kodlayıcılar, üretken çekişmeli ağlar (GAN'ler) vb. gibi farklı görüntü modeli türleri vardır. Her model türünün, göreve ve verilere bağlı olarak kendi avantajları ve dezavantajları vardır.

Bu çalışmada Inception-v3 ve ResNet-v2(50) pre-trained modelleri kullanarak image embedding'ler elde edilmiştir. Image Embedding için kullanılabilecek tüm pretrained

modellere Kaggle Models (TF Hub) altından Task filtresini “Image Future Vector” kullanarak ulaşabilirsiniz.

Inception v3

Inception v3, Google tarafından geliştirilen Inception ailesinin bir üyesidir ve özellikle görüntü sınıflandırma ve tanıma görevlerinde kullanılır. Bu model, Inception v1 (GoogLeNet olarak da bilinir) ve Inception v2'nin geliştirilmiş bir sürümüdür.

ResNet v2 (Residual Networks Version 2)

ResNet, Microsoft Research tarafından geliştirilmiş ve “artık öğrenme” (residual learning) yaklaşımını kullanan bir ağıdır. ResNet v2, orijinal ResNet modelinin (ResNet v1) geliştirilmiş bir versiyonudur.

Her iki model de derin öğrenme topluluğu içinde önemli kilometre taşlarıdır ve modern görüntü işleme sistemlerinin gelişiminde büyük rol oynamışlardır.

Geliştirdiğimiz kod parçasında input yani film görselleri “data/input/images” altındaki film görsellerini input olarak almakta ve ardından paralel olarak 4 thread’de image to array ardından da pretrained modellere sırayla girerek embedding vektörleri oluşmaktadır. Embedding vektörlere data/processed/{inception_v3, resnet_v2}.csv.gz dosyalarından ulaşabilirsiniz.

İlgili kaynak koduna proje dizin içerisinde “generate_image_embeddings.py” üzerinden ulaşabilirsiniz.

Veri Son İşleme

Bu kısımda Python ve IDE olarak Jupyter Notebook kullanarak Film Görseller ile elde ettiğimiz Image Embeddingleri üzerinde MultiClass Classification modellemesi öncesinde bazı post process işlemleri gerçekleştirdik. Aşağıda tüm yapılanlara DataPostProcess.ipynb üzerinden ulaşabilirsiniz. Bunlardan bazıları;

1. Elimizde PreProcess adımıyla elde ettiğimiz df_master_data (811 x 9) veri seti ile ImageEmbedding adımıyla elde ettiğimiz veri setleri [ImgEmbeddingwithInception-v3 (997 x 2049) ve ImgEmbeddingwithResNet-v2 (997 x 2049)] ile ImgName (ya da ImgId) üzerinden birleştirme işlemi yapmaktayız.
2. Birleştirme işlemi sonrasında ve modelleme için gerekli olan featureların seçilmesi ile birlikte elimizde “0, 2, 3, ... 2047, Genre” kolonlarından oluşan 2 ayrı veri setlerimiz bulunmaktadır. Veri setlerimize “data/result/{ImgEmbeddingwithInception-v3, ImgEmbeddingwithResNet-v2} buradan ulaşabilirsiniz.

Bu projenin ilk aşamalarında, sonraki model eğitimi ve değerlendirmesinin temelini oluşturmak için titiz veri ön işleme çok önemliydi. Görüntü tanımlayıcıları, adlar, başlıklar ve ilişkili türler gibi film afişleri hakkında temel bilgileri içeren veri kümesi bir CSV dosyasından alınmış ve Pandas kullanılarak yüklenmiştir. Bu aşamanın kritik bir yönü, kategorik film türlerinin bir LabelEncoder kullanılarak kodlanmasını içeriyordu. Bu dönüşüm, kategorik

verileri sayısal bir formata dönüştürdü; bu, modelin çeşitli film türlerini anlamasını ve sınıflandırmasını kolaylaştırmak için gerekli bir adımdı. Süreç, belirli bir klasör konumunda depolanan film afişi görüntülerinden oluşan 'SampleMoviePosters' veri kümesinde bulunan görüntü verilerini incelemiştir. Ön işleme hattı birkaç temel adımı kapsamaktadır:

Görüntü Yükleme ve Ön İşleme:

Kod, belirlenen klasördeki görüntülerin varlığını kontrol etti ve Python Imaging Library (PIL) kullanarak bunları yükledi. Görüntüler, derin öğrenme modelinde kullanılmaları için gerekli olan yeniden boyutlandırma ve tensör formatına dönüştürme dahil olmak üzere temel dönüşümlerden geçirildi.

Veri Kümesi Bölme:

sklearn.model_selection modülündeki train_test_split işlevi kullanılarak veri kümesi verimli bir şekilde eğitim ve test alt kümelerine bölünmüştür. 80-20 oranında yapılandırılan bu bölünme, model eğitimi ve değerlendirmesi arasında bir denge sağladı. İşlenen verilerin PyTorch çerçevesine entegrasyonunun merkezinde özel bir veri kümesi sınıfı olan 'CustomDataset'in oluşturulması vardı. Bu sınıf, sonraki derin öğrenme işlem hattına sorunsuz entegrasyon için veri kümesinin düzenlenmesinde ve yapılandırılmasında çok önemli bir rol oynamıştır. 'CustomDataset' sınıfı, işlenmiş görüntü verilerini ('X') ve bunlara karşılık gelen kodlanmış etiketleri ('y') kapsüllemek üzere tasarlanmıştır. Veri kümesini sağlanan girdilerle başlatarak ve netlik için veri kümesinin insan tarafından okunabilir bir temsili sunarak, kurucusu içinde temel işlevleri bir araya getirmiştir. __len__ ve __getitem__ gibi geçersiz kılınan yöntemler, toplam örnek sayısını sağlayarak ve belirli özellikleri (görüntü verileri) ilgili hedef etiketleriyle birlikte alarak veri kümesi içindeki tek tek örneklerle kolay erişimi kolaylaştıracak şekilde yazıldı.

Tensör Dönüşümü ve DataLoader Kurulumu:

Veri kümesini transforms.ToTensor() kullanarak PyTorch tensörlerine dönüştürmek, PyTorch ekosistemi içinde uyumluluk için çok önemliydi. Daha sonra, model eğitimi ve test aşamaları sırasında verimli toplu işlemeye olanak tanıyan DataLoader örnekleri oluşturuldu. Bu uygulama, veri setinin bir şekilde hazırlanmasını ve yapılandırılmasını sağlamış, model eğitimi ve değerlendirmesinin sonraki aşamalarına sorunsuz entegrasyon için hazır hale getirilmiş, sağlam ve etkili derin öğrenme modeli performansı için zemin hazırlamıştır.

ResNet50 Mimarisi ve Önceden Eğitilmiş Katmanlar:

Artık bağlantılara sahip 50 katmandan oluşan ResNet50 mimarisi, derinliği ve görüntü sınıflandırma görevlerinde kanıtlanmış performansı nedeniyle seçilmiştir. ImageNet'ten önceden eğitilmiş katmanlardan yararlanan ResNet50, karmaşık desenleri ve temsilleri yakalayarak film afişi görüntülerinden özellik çıkarımını kolaylaştırdı. İnce Ayar için Son Katmanların Dondurulması adına, önceden eğitilmiş ResNet50 modelinin son katmanları çözülerek daha önce öğrenilen genel özellikler korunurken film afişi veri kümesinin özelliklerine uyum sağlandı. Bu strateji, ön eğitimden elde edilen temel bilgilerden ödün vermeden modelin alana özgü özellikleri anlamasını geliştirmeyi amaçlamıştır. Adam

optimize edici, çapraz entropi kayıp fonksiyonu ile birleştğinde model eğitimi için dengeli bir yaklaşım sağlamıştır. Adam'ın uyarlanabilir öğrenme oranları yakınsamayı hızlandırırken, çapraz entropi kaybı tahmin edilen ve gerçek türler arasındaki sınıflandırma tutarsızlıklarının etkili bir şekilde hesaplanmasını kolaylaştırmıştır. Eğitimde Kullanılan hiper parametreler istikrarlı bir eğitim süreci sağlamak amacıyla sistemin karmaşıklığı göz önünde bulundurularak seçilmiştir: Hızlı yakınsama ve model kararlılığı arasında bir denge kurmak için öğrenme oranı, 0,001 olarak ayarlanmıştır. Yığın boyutu eğitim iterasyonları sırasında veri yığınlarını verimli bir şekilde işlemek için 32 olarak yapılandırılmıştır. epochs eğitim 100 epoch üzerinden gerçekleştirilerek modelin kapsamlı bir ince ayar yapmadan veri kümesinin nüanslarına kademeli olarak uyum sağlamasına olanak tanınmıştır.

Eğitim Süreci ve Sonuçları

Eğitim süreci, film afişi görüntülerinin çeşitli temsillerinden öğrenmesine ve genelleştirmesine olanak tanıyan artırılmış veri gruplarının modele beslenmesini içeriyordu. Artırma teknikleri, özellikle ayrıntılı olmasa da daha iyi model genellemesi için veri kümesi çeşitliliğini artırmayı amaçlamıştır, sistemin karmaşıklığı nedeniyle proje, belirli doğruluk ölçütlerinden ziyade model yakınsaması ve kararlılığına odaklanmıştır. Epochlar boyunca kayıp eğrilerinin izlenmesi, modelin sınırlı veri kümesine aşırı uyum sağlamadan ilgili kalıpları öğrenmesini sağlayarak yakınsama konusunda içgörü sağlamıştır.

Sistemin karmaşıklığı göz önüne alındığında, proje belirli performans ölçütleri üzerinde durmamıştır. Her ne kadar eğitim ve test süreçlerinde kesinlik birincil odak noktası olsa da istikrarlı bir öğrenme sürecini sürdürmek ve doğruluk temelli değerlendirmelere girmeden modelin film afişi veri kümesine uyarlanabilirliğini sağlamak olmuştur. Proje kapsamında, Google Colab'daki GPU kaynaklarının kullanımı eğitim sürecini önemli ölçüde hızlandırmış ve bilgisayarla görme görevlerinde derin öğrenme metodolojilerinin keşfedilmesini kolaylaştırmıştır. Poster görüntülerine dayalı olarak film türlerini sınıflandırmada önceden eğitilmiş modellerden, ResNet50'den yararlanmanın etkili olabileceği gösterilmiştir. Transfer öğrenmenin uygulanması ve son katmanların ince ayarının yapılması, modelin veri kümesinin nüanslarına uyum sağlamasına olanak tanıyarak sınıflandırma görevleri için karmaşık özellikler çıkarma yeteneğini sergilemiştir.

Resnet50 bağlamında proje, derin öğrenme tekniklerinin bilgisayarla görme görevlerinde, özellikle de poster görüntülerine dayalı film türü sınıflandırmasında pratik olarak uygulanmasına ilişkin değerli bilgiler sağlamıştır. Ayrıca elde edilen özellik vektörlerinin derin sinir ağı modeli kurulup burada başarının ölçülmesi de elde edilen embeddinglerin farklı bir kullanım opsiyonunu da sunmuştur. Özellik vektörlerinin derin sinir ağı modelinde yüksek başarı elde edildiği söylenemez. Odak noktası en yüksek doğruluğa ulaşmaktan ziyade model uygulaması ve kararlılık olsa da proje bu alanda gelecekteki keşif ve iyileştirmeler için sağlam bir temel oluşturmuştur. Google Colab'da GPU'ların kullanılması ve süreçlerin gözlenmesi, karmaşık derin öğrenme projelerinin üstlenilmesinde erişilebilir hesaplama kaynaklarının öneminin altını çizdi ve platformun önemli hesaplama gücüne ücretsiz erişim sağlayarak yapay zeka araştırma ve geliştirmesini rolünü vurguladı. Genel olarak proje, görüntü sınıflandırma görevlerinde derin öğrenme modellerinin etkinliğini ve bu tür çabaları

kolaylařtırmada erişilebilir GPU kaynaklarının önemini gösteren değerli bir öğrenme deneyimi sundu.

Projeye ait tüm kodları, sonuçları, grafikleri ve daha fazlasını ilgili adreste görebilirsiniz:

https://github.com/metinuslu/blm5103_cv_project

Anıl Özcan

Hamza Cantürk

Metin Uslu

235B7022

235B4005

235B7014