

Makine Öğrenmesi ile Mimari Akımların Sınıflandırılması

Architectural Style Classification with Machine Learning

Hacer Sueda Efe - 211307060

Bilişim Sistemleri Mühendisliği
Teknoloji Fakültesi - Kocaeli Üniversitesi
İzmit, Kocaeli /Türkiye
suedaefe@gmail.com

[Google Drive](#)

I. GİRİŞ

Abstract— This report contains information about the process of collecting visual data from Google using an image scraper with Selenium library and Python, and then training and testing 5 different deep learning models with this collected data using the Hugging Face Transformers library with Python again. The main goal of the study is to test the collected data using different deep learning models and to evaluate the performance of these models. For this purpose, a methodology that includes the data collection process, model training and analysis of the results was followed. The subject of the study is to identify different architectural styles, 4 styles were selected, these are Art Nouveau, Baroque, Gothic and Islamic architectures. Machine learning is an extremely flexible technology that can be used in a wide range of fields and it is thought that it can play an important role in the field of architecture as well.

Özet— Bu rapor Selenium ve Python kullanılarak yazılan bir image scraper ile Google'dan görsel veri toplama ve ardından yine Python ile Hugging Face Transformers kütüphanesini kullanarak 5 farklı derin öğrenme modelini bu toplanan verilerle eğitime ve test etme sürecini içermektedir. Çalışmanın ana hedefi farklı derin öğrenme modellerini kullanarak toplanılan verileri test etmek ve bu modellerin performansını değerlendirmektir. Bu amaç doğrultusunda, veri toplama süreci, model eğitimi ve sonuçların analizi adımlarını içeren bir metodoloji izlenmiştir. Çalışmanın konusu farklı mimari akımların tespit edilmesi üzerinedir, 4 adet akım seçilmiştir bunlar Art Nouveau, Barok, Gotik ve İslam mimarileridir. Makine öğrenmesi, geniş yelpazedeki alanlarda kullanılabilecek son derece esnek bir teknoloji olup, mimari alanında da önemli bir rol oynayabileceği düşünülmektedir.

Keywords— Machine Learning, Architectural Styles, Selenium, Transformer Models, K-Fold Cross Validation, Python

Makine öğrenmesi günümüzde birçok farklı alanda kullanılan ve hızla popülerleşen bir teknolojidir. Bu projeyle de bu teknolojiyi mimari alanda da faydalı olabilecek bir hale getirmek amaçlanmıştır. Yapılan araştırmalar sonucunda literatürde bu alanda pek fazla çalışma yapılmadığı görülmüş bu sebeple bu konunun faydalı olabileceği düşünülmüştür.

Mimari yapılar tarih boyunca milletlerin sosyal ve kültürel yapısını, estetik anlayışlarını ve o döneme dair daha pek çok durumu yansıtmışlardır, bu açıdan önemli izler taşımaktadırlar, farklı akımların da birbirinden ayrılması bu bağlamda önem arz etmektedir. Bu projede Art Nouveau, Barok, Gotik ve İslam mimarisi gibi önemli dört akım seçilerek bu akımların belirgin özelliklerini makine öğrenimi modellerinin tespit edebilme yeteneği araştırılmıştır.

II. VERİ TOPLAMA (WEB SCRAPING)

Google'dan görsel veri toplama sürecinde python ile bir data scraper kodlanmıştır, Selenium kütüphanesi tercih edilmiştir. Yazılan scraper ChromeDriver kullanarak yazılan anahtar kelimeleri Google görsellerde aramakta sayfayı sonuna kadar kaydırdıktan sonra çıkan resimlere sırayla tıklayıp resimleri büyütme, büyük olan resimlerin URL'lerini toplamaktadır. Bu URL'ler saklanmıştır. Belirli bir sayıda URL toplandıktan sonra, fonksiyon bu URL setini döndürmektedir ve görselleri belirtilen dosya yoluna indirmektedir. İlk başta resimleri büyütmeden direkt olarak kapak resminden url toplama işlemi yapılmış fakat bu küçük boyutlu resimleri indirildiği için ve bunlar kalitesiz olduğu için başta anlatılan metoda geçilerek daha iyi sonuç verdiği gözlemlenmiştir. Anahtar kelimeler verilirken başlangıçta sadece akımların adı yazılarak (örneğin barok mimarisi) veri toplandığı zaman pek çeşitli veri elde edilememiş daha detaylı ve kapsamlı olması açısından ön bir araştırma yapılarak her mimari akımdan yaklaşık 20-25 adet yapı bulunmuş ve spesifik olarak tek tek bu

yapıların isimleri ve yanlarına ön cephe, yan cephe gibi ek kelimeler ile arama yapılarak daha geniş bir veri seti ele edilmiştir. Türkçe ve İngilizce olarak arama yapılmıştır. Bu veri toplama sürecinde herhangi bir blokaja takılmamak için resimleri indirirken aralara random bekleme süreleri eklenmiştir.

Manuel olarak veri temizleme işleminin ardından ham olarak sınıf başı yaklaşık 400 resim geriye kalmış bu resimler internette hazır toollar kullanılarak, grey scale, sağ sol 180 derece rotasyon gibi işlemlere tabi tutularak sınıf başı 1800'e kadar çoğaltılmıştır, tüm resimler 224x224 olarak boyutlandırılmıştır. Class imbalance problemi olmaması için her sınıfa eşit sayıda veri atanmıştır, toplam sonuç olarak 7200'lük bir data seti elde edilmiştir.

III. MODEL EĞİTİMİ VE PERFORMANSI

A. Kodun Çalışma Prensipleri

Bu aşamada PyTorch, Transformers, Sklearn, Numpy, Matplotlib gibi kütüphaneler model eğitimi, değerlendirmesi, metrik hesaplaması, grafiklerin çizilmesi gibi işlemler için kullanılmıştır.

Veri hazırlama sürecinde görüntüler belirtilen dosya yolundan erişilerek eğitim ve test süreçleri için çeşitli dönüşümlere tabi tutulmuştur. Eğitim verileri renk oynamaları, rastgele döndürme ve kırma gibi farklı tekniklerle tekrar çeşitlendirilmiştir. Bu süreç modelin farklı ve çeşitlendirilmiş veri senaryolarında daha iyi genelleştirme yapabilmesini sağlamak için uygulanmıştır.

Modelin sınıflandırma katmanına dropout eklenmiştir. Dropout, modelin ezberlemesini (overfit) önlemek için kullanılan bir regülasyon yöntemidir ve modelin genelleme yeteneğini artırmaya yardımcı olur.

Eğitim sürecinde görüntü ve etiketler GPU'ya taşınarak işlenmektedir. Modelin çıktıları ve gerçek etiketler arasındaki kayıp hesaplanmakta, backpropagation yöntemiyle modelin ağırlıkları güncellenmektedir. Gradients accumulation kullanılarak belirli bir adım sayısı sonra optimizasyon yapılmıştır, bu yöntem, büyük veri setleri üzerinde eğitim yaparken bellek tasarrufu sağlamaktadır.

Modelin eğitimi sırasında kullanılan önemli hiperparametrelerden olan learning rate ve weight decay optimizasyon sürecinin kritik bileşenleridir. Bu projede AdamW optimizasyon algoritması kullanılmıştır. AdamW, weight decay'i doğrudan optimize eden bir algoritmadır. Bu parametreler modelin ağırlıklarının güncellenme hızını ve düzenliliğini kontrol ederek modelin ezberlemesi problemini önlemeye yardımcı olur. StepLR öğrenme oranı düşürücüsü de kullanılarak öğrenme oranı her epoch sonunda yarıya düşürülmüştür (gamma=0.5). Bu adım, modelin eğitim ilerledikçe daha

küçük adımlarla güncellenmesini sağlar, böylece daha ince ayarlar yapılır ve öğrenme süreci daha kararlı hale gelir.

Değerlendirme sürecinde, modelin performansı test verileri üzerinde ölçülmektedir. Test süreci sırasında, loss ve accuracy değerleri hesaplanmaktadır. Ayrıca, ROC eğrisi için gerekli olan tüm etiketler ve tahmin olasılıkları kaydedilir. Bu, modelin her bir sınıf için ne kadar iyi performans gösterdiğini anlamak içindir.

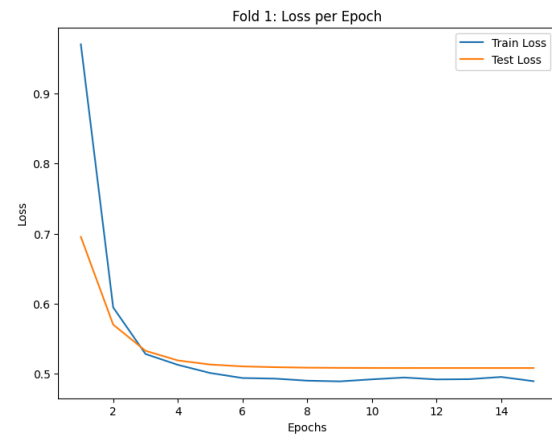
Kodun en önemli kısmı, kullanılan k-fold cross-validation yöntemidir. Bu yöntem, veri setini k eşit parçaya böler ve her bir parça bir kez test verisi olarak kullanılırken diğer k-1 parça eğitim verisi olarak kullanılır. Böylece modelin genelleme yeteneği daha iyi değerlendirilir. Bu projede k = 5 olarak alınmıştır yani veri seti %80 eğitim %20 test şeklinde bölünmüştür. Eğitim ve testin loss accuracy'leri her epoch için yazdırılmış, epoch loss grafikleri de her fold için çizdirilmiştir. Ayrıca, her bir sınıf için precision, recall, specificity, f1-score, accuracy, MCC ve AUC gibi metrikler hesaplanarak fold başı yazdırılmıştır. 5 foldun sonunda ortalama epoch loss ve accuracy grafikleri ve bir ROC eğrisi sunulmuştur. Bu grafikler modelin genel performansını görselleştirmiştir.

B. Kullanılan Modeller ve Sonuçlar

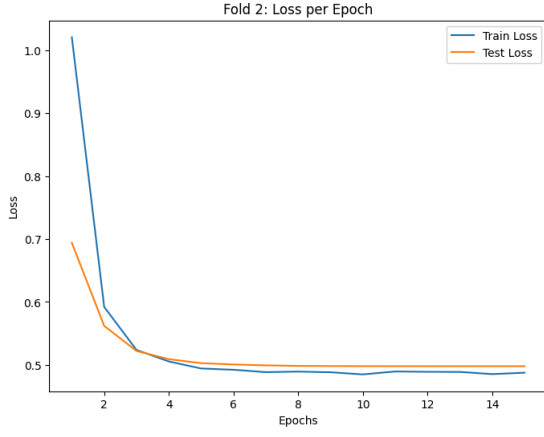
1) Google ViT :

Google tarafından geliştirilen ViT, görüntü sınıflandırma için transformer mimarisini kullanan bir modeldir. Görüntüleri küçük parçalara böler ve her parçayı bir diziye dönüştürerek işlem yapar. Bu model, özellikle büyük veri setlerinde etkili sonuçlar vermektedir ve derin öğrenme alanında yeni bir yaklaşım olarak kabul edilmektedir. Projede kullanılan versiyonu google/vit-base-patch16-224 dır.

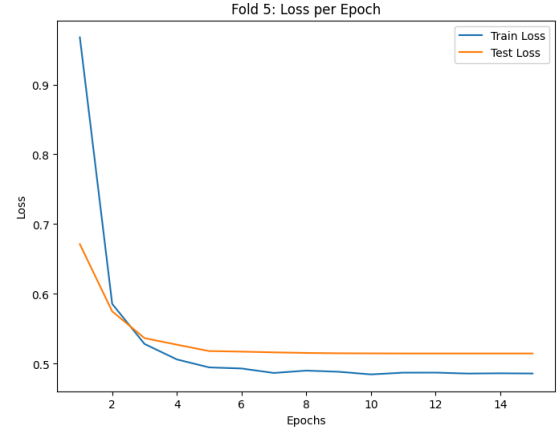
Verilen değerler dropout = 0.5 learning rate = 0.000005, weight decay = 0.01 batch size = 8 şeklindedir.



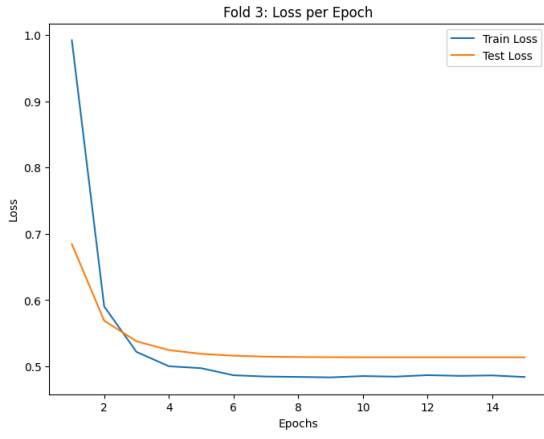
Şekil 1: Google ViT Epoch Loss Fold 1



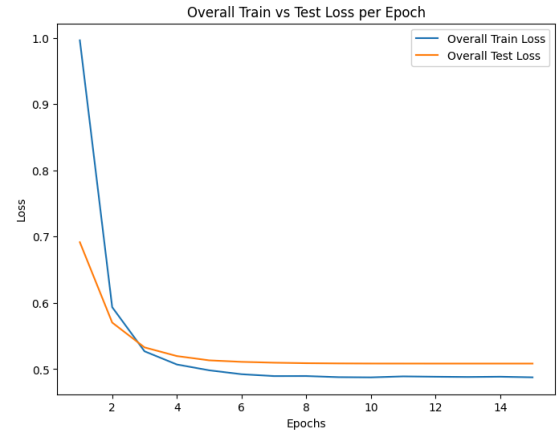
Şekil 2: Google ViT Epoch Loss Fold 2



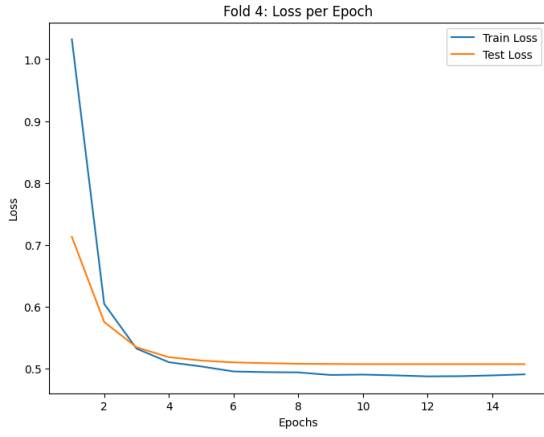
Şekil 5: Google ViT Epoch Loss Fold 5



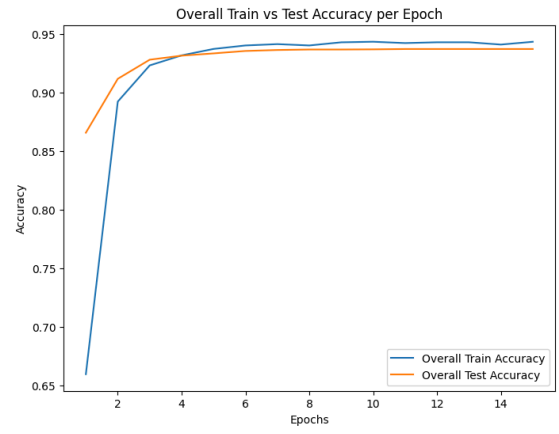
Şekil 3: Google ViT Epoch Loss Fold 3



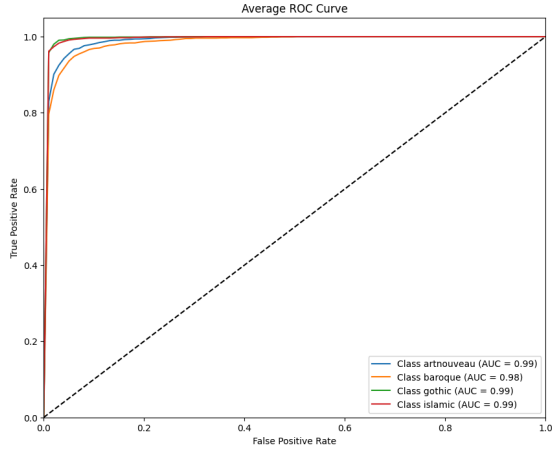
Şekil 6: Google ViT Overall Epoch Loss Grafiği



Şekil 4: Google ViT Epoch Loss Fold 4



Şekil 7: Google ViT Accuracy Grafiği



Şekil 8: Google ViT ROC Grafiği

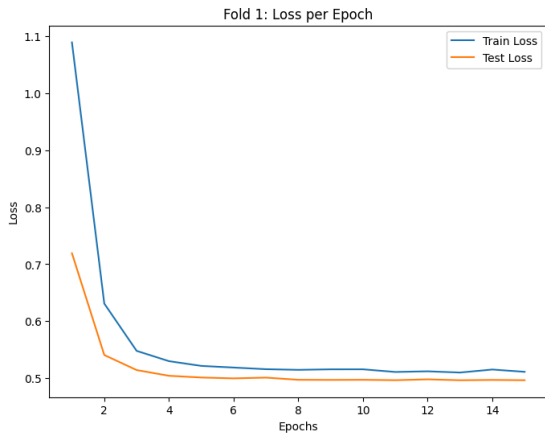
	ArtNouveau	Baroque	Gothic	Islamic	Overall
Precision	0.909620	0.908940	0.961640	0.971600	0.937950
Recall	0.927100	0.896680	0.967940	0.957460	0.937295
F1-score	0.917880	0.902120	0.964720	0.964420	0.937285
Accuracy	0.958400	0.951540	0.982320	0.982300	0.968640
Specificity	0.968880	0.969820	0.987120	0.990600	0.979105
MCC	0.890480	0.870500	0.953000	0.952740	0.916680
AUC	0.991500	0.986580	0.997840	0.997820	0.993435

Şekil 9: Google ViT Sonuç Metrikleri

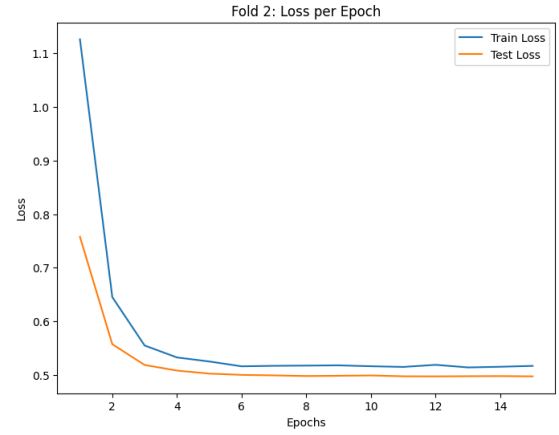
2) Microsoft Swin :

Microsoft tarafından geliştirilen Swin, görüntü işleme görevleri için optimize edilmiş bir transformer modelidir. Swin görüntüleri farklı boyutlardaki pencerelere bölerek işlem yapar. Bu yöntem, hem yerel hem de küresel bilgiye odaklanarak yüksek performans sağlar ve çeşitli bilgisayarla görme görevlerinde üstün başarı gösterir. Projede kullanılan versiyonu microsoft/swin-base-patch4-window7-224 dür.

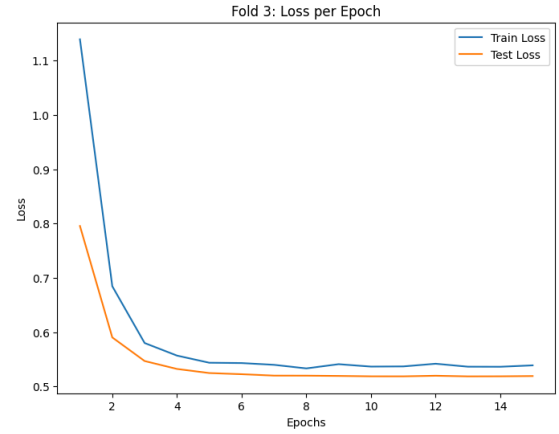
Verilen değerler dropout = 0.3 learning rate = 0.000005 weight decay = 0.01 batch size = 16 şeklindedir.



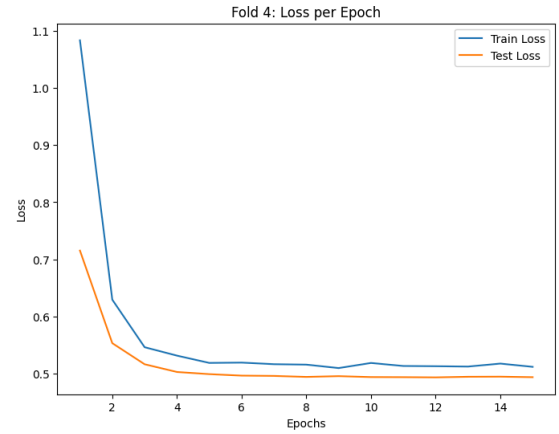
Şekil 10: Microsoft Swin Epoch Loss Fold 1



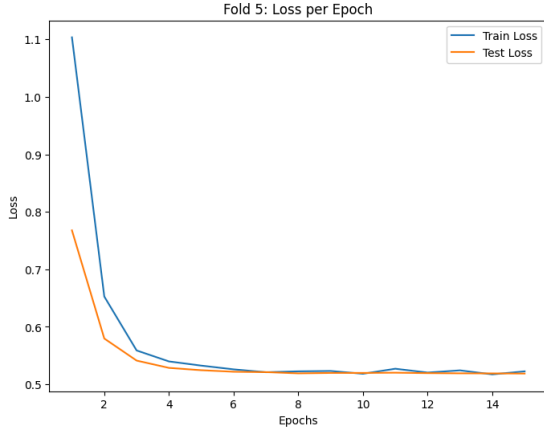
Şekil 11: Microsoft Swin Epoch Loss Fold 2



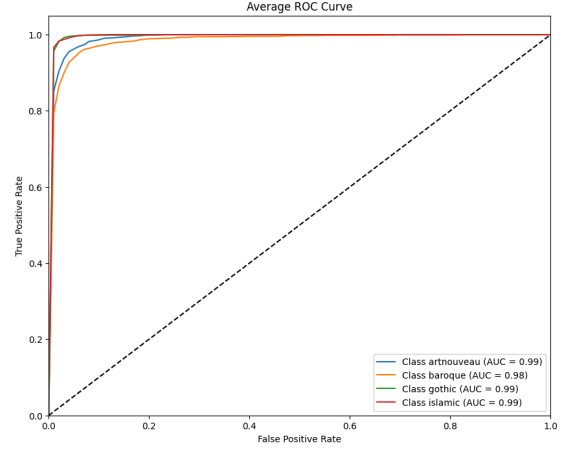
Şekil 12: Microsoft Swin Epoch Loss Fold 3



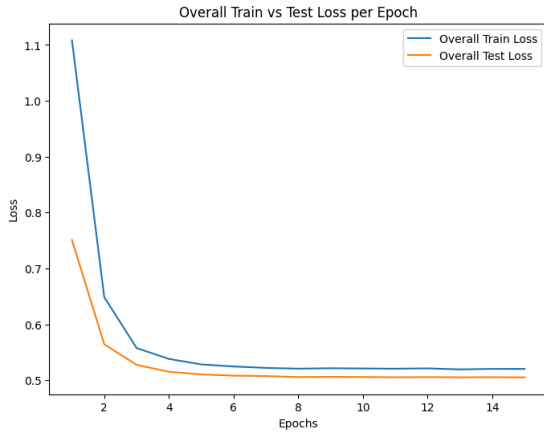
Şekil 13: Microsoft Swin Epoch Loss Fold 4



Şekil 14: Microsoft Swin Epoch Loss Fold 5



Şekil 17: Microsoft Swin ROC Grafiği



Şekil 15: Microsoft Swin Overall Epoch Loss Grafiği

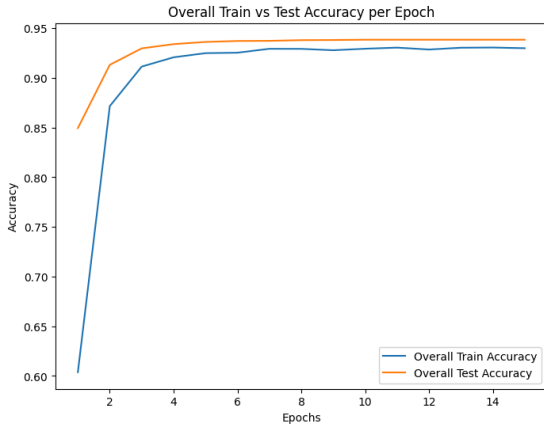
	ArtNouveau	Baroque	Gothic	Islamic	Overall
Precision	0.935820	0.896940	0.953160	0.969120	0.938760
Recall	0.900560	0.915460	0.974040	0.963520	0.938395
F1-score	0.917700	0.906000	0.963380	0.966220	0.938325
Accuracy	0.959680	0.952500	0.981480	0.983120	0.969195
Specificity	0.979380	0.964800	0.983960	0.989680	0.979455
MCC	0.891440	0.874380	0.951220	0.955100	0.918035
AUC	0.993020	0.986480	0.998000	0.998720	0.994055

Şekil 18: Microsoft Swin Sonuç Metrikleri

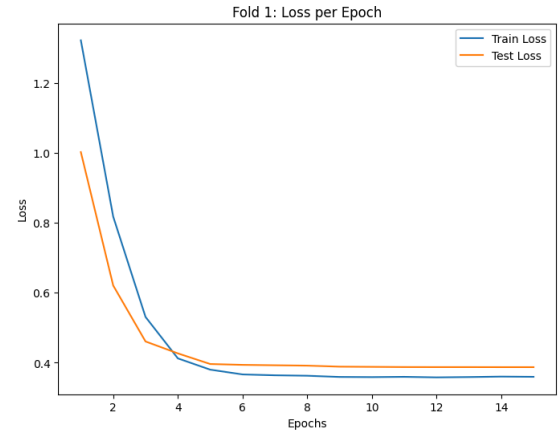
3) Facebook DeiT :

Facebook tarafından geliştirilen DeiT veri verimliliği yüksek bir görüntü sınıflandırma modelidir. DeiT daha az veriyle yüksek performans elde etmek için optimize edilmiştir. Bu model diğer yöntemlere kıyasla daha az veri ile eğitim yaparak benzer veya daha iyi sonuçlar elde eder, bu da DeiT'i veri setinin sınırlı olduğu durumlarda iyi bir seçim haline getirir. Bu projede kullanılan versiyonu facebook/deit-base-distilled-patch16-224 dır.

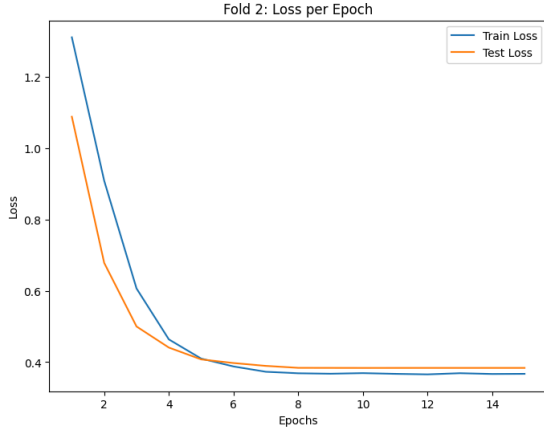
Verilen değerler dropout = 0.3 learning rate = 0.0005, weight decay = 0.01 batch size = 8 şeklindedir.



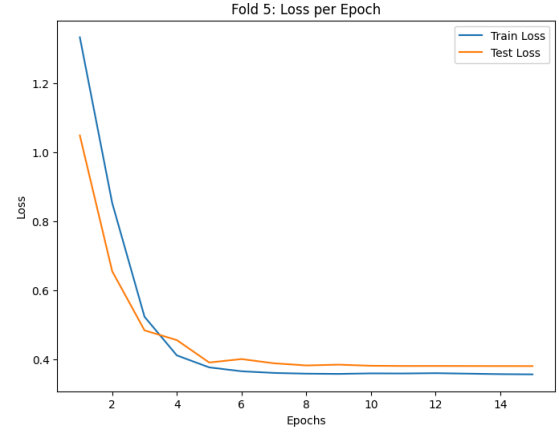
Şekil 16: Microsoft Swin Accuracy Grafiği



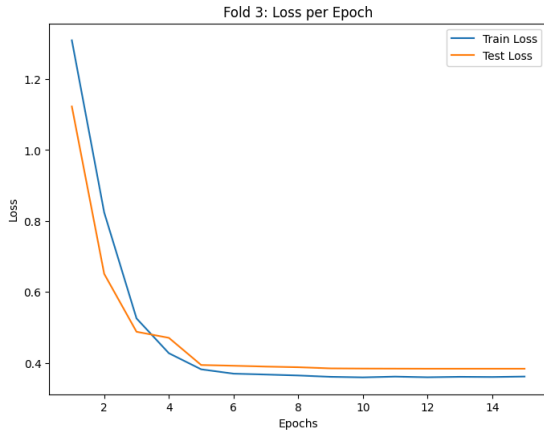
Şekil 19: Facebook DeiT Epoch Loss Fold 1



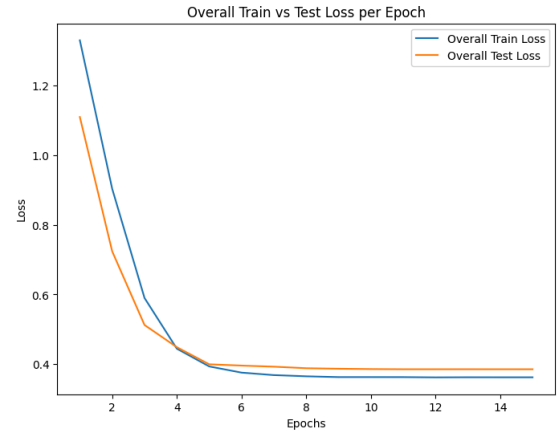
Şekil 20: Facebook DeiT Epoch Loss Fold 2



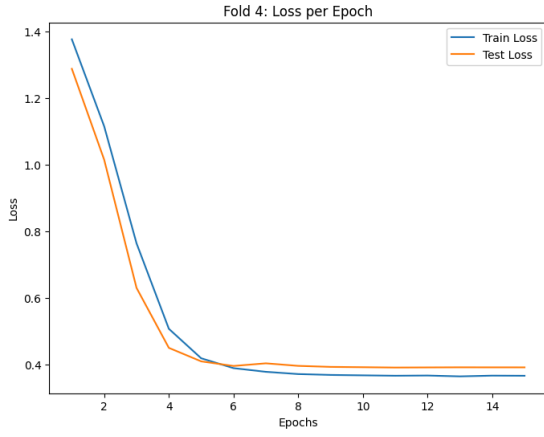
Şekil 23: Facebook DeiT Epoch Loss Fold 5



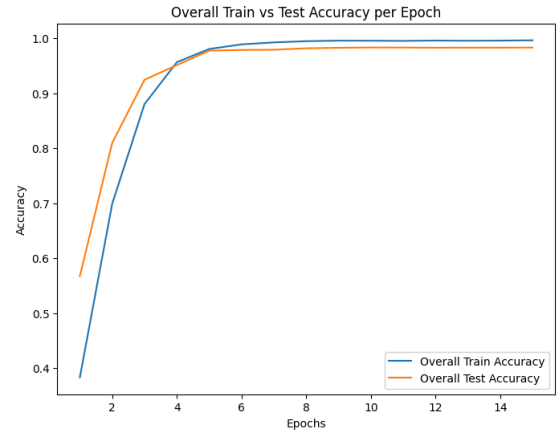
Şekil 21: Facebook DeiT Epoch Loss Fold 3



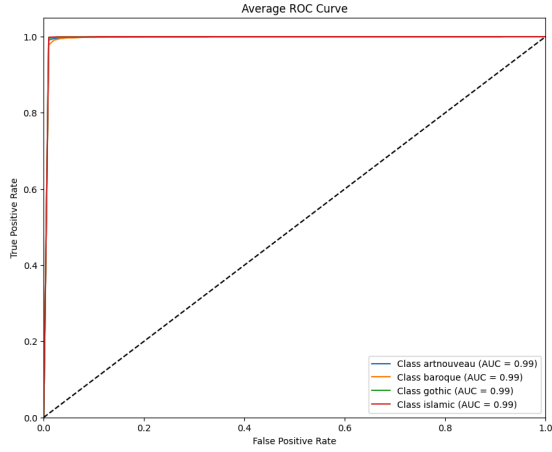
Şekil 24: Facebook DeiT Overall Epoch Loss Grafiği



Şekil 22: Facebook DeiT Epoch Loss Fold 4



Şekil 25: Facebook DeiT Accuracy Grafiği



Şekil 26: Facebook DeiT ROC Grafiği

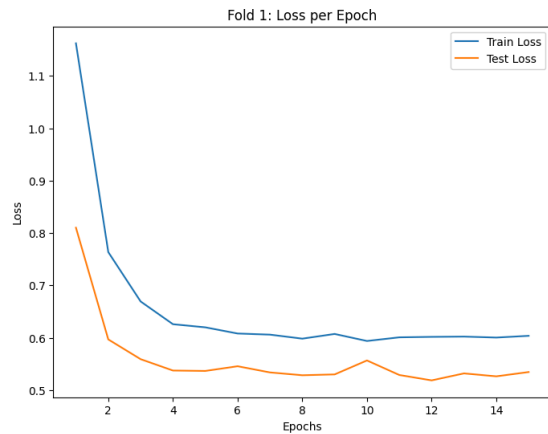
	ArtNouveau	Baroque	Gothic	Islamic	Overall
Precision	0.982960	0.980980	0.976260	0.991740	0.982985
Recall	0.982900	0.967980	0.995600	0.985100	0.982895
F1-score	0.982880	0.974380	0.985800	0.988360	0.982855
Accuracy	0.991440	0.987300	0.992820	0.994220	0.991445
Specificity	0.994280	0.993760	0.991900	0.997240	0.994295
MCC	0.977200	0.966000	0.981080	0.984540	0.977205
AUC	0.998760	0.998860	0.999500	0.999780	0.999225

Şekil 27: Facebook DeiT Epoch Sonuç Metrikleri

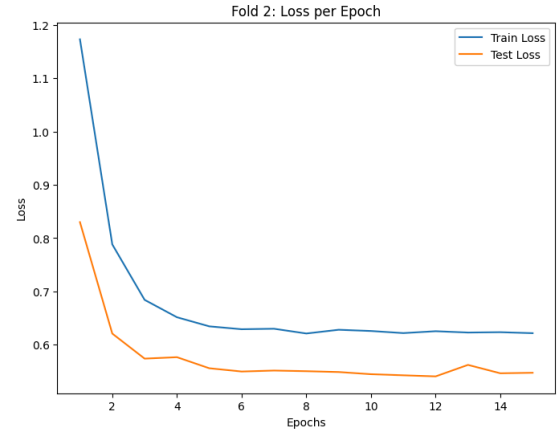
4) Facebook LeViT :

LeViT, Facebook tarafından geliştirilen bir görüntü sınıflandırma modelidir. LeViT transformer ve CNN'lerin avantajlarını birleştirerek düşük hesaplama maliyetleri ile yüksek doğruluk sağlamayı hedefler. Özellikle mobil gibi kaynakların sınırlı olduğu ortamlarda kullanım için uygundur. Bu projede kullanılan versiyonu facebook/levit-384 dür.

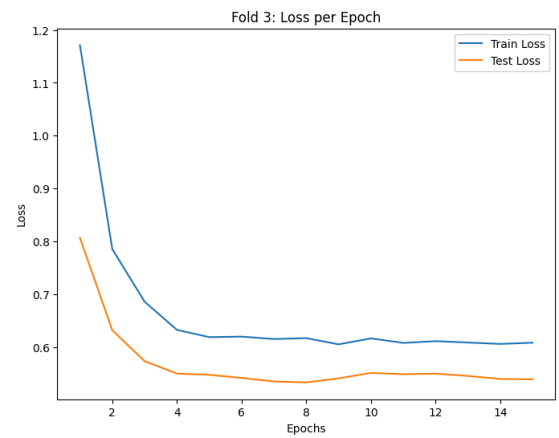
Verilen değerler dropout = 0.6 learning rate = 0.0001, weight decay = 0.01 batch size = 16 şeklindedir.



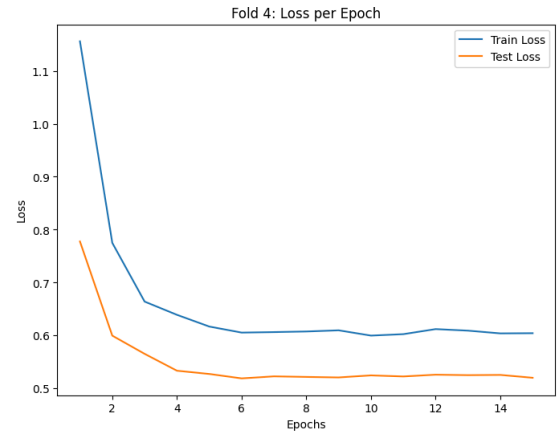
Şekil 28: Facebook LeViT Epoch Loss Fold 1



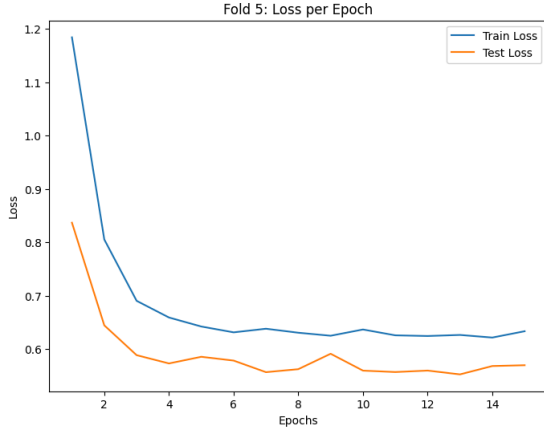
Şekil 29: Facebook LeViT Epoch Loss Fold 2



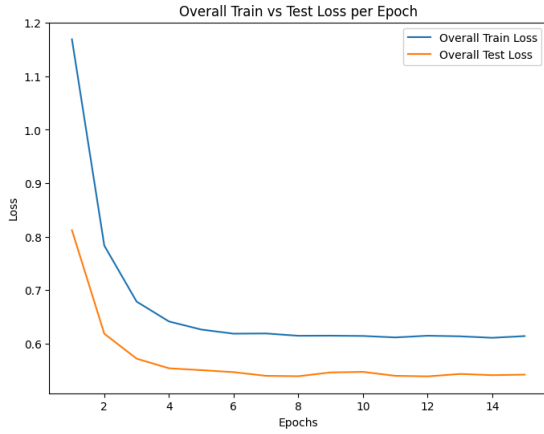
Şekil 30: Facebook LeViT Epoch Loss Fold 3



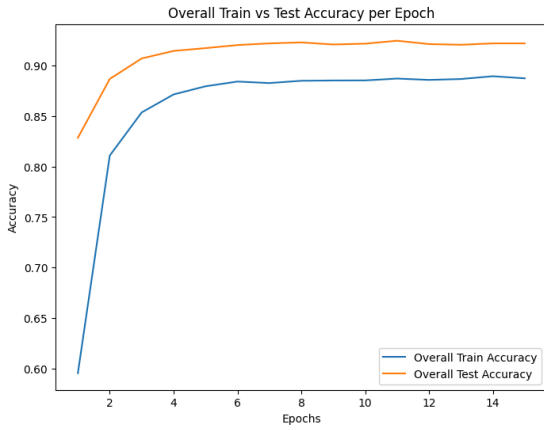
Şekil 31: Facebook LeViT Epoch Loss Fold 4



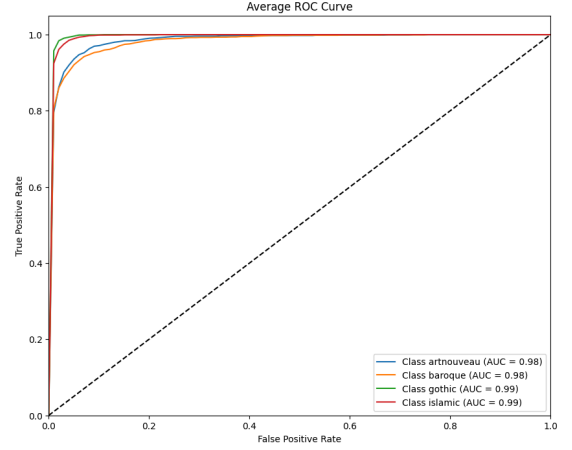
Şekil 32: Facebook LeViT Epoch Loss Fold 5



Şekil 33: Facebook LeViT Overall Epoch Loss Grafiği



Şekil 34: Facebook LeViT Accuracy Grafiği



Şekil 35: Facebook LeViT ROC Grafiği

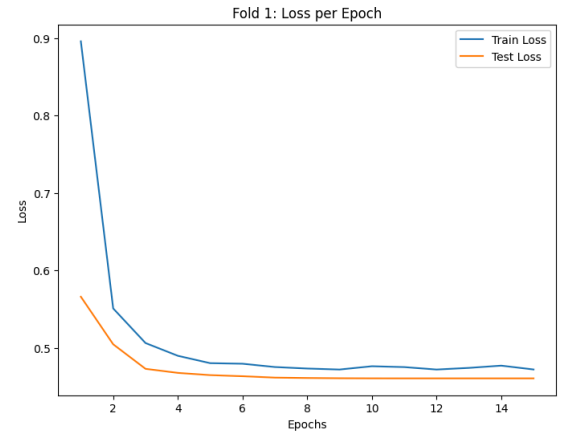
	ArtNouveau	Baroque	Gothic	Islamic	Overall
Precision	0.933540	0.875480	0.918380	0.969000	0.924100
Recall	0.856900	0.920460	0.986780	0.923780	0.921980
F1-score	0.893060	0.897020	0.951300	0.945240	0.921655
Accuracy	0.948880	0.946960	0.974720	0.973340	0.960975
Specificity	0.979560	0.955780	0.970720	0.989860	0.973980
MCC	0.861380	0.862180	0.935400	0.928620	0.896895
AUC	0.987460	0.984000	0.998520	0.996980	0.991740

Şekil 36: Facebook LeViT Sonuç Metrikleri

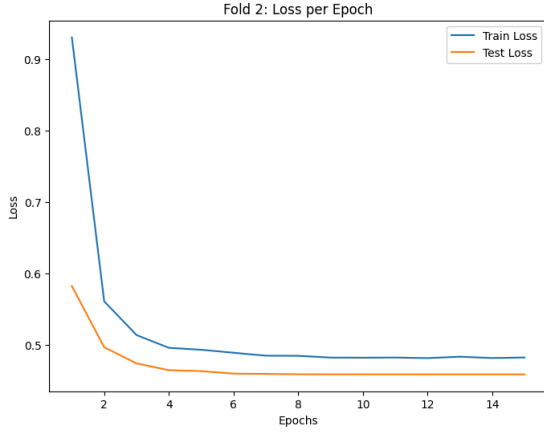
5) Microsoft BeiT :

BEiT, Microsoft tarafından geliştirilen bir görüntü sınıflandırma modelidir. BEiT görüntüleri küçük parçalara ayırarak bu parçalar üzerinde transformer mimarisi ile işlem yapar. Bu model, ön eğitim sırasında büyük veri setlerinden öğrenerek, ince ayar yapıldığında yüksek performans gösterir. Bu projede kullanılan versiyonu microsoft/beit-base-patch16-224 dür.

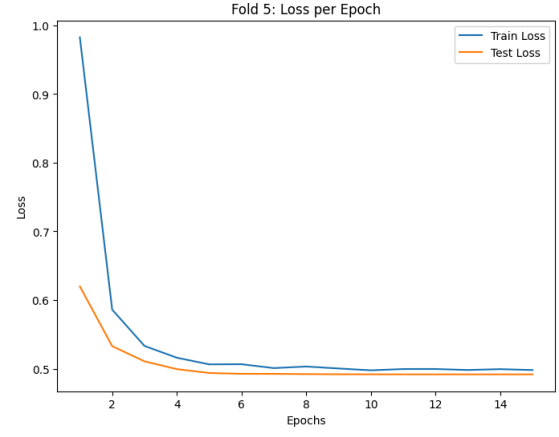
Verilen değerler dropout = 0.5 learning rate = 0.000005, weight decay = 0.01 batch size = 16 şeklindedir.



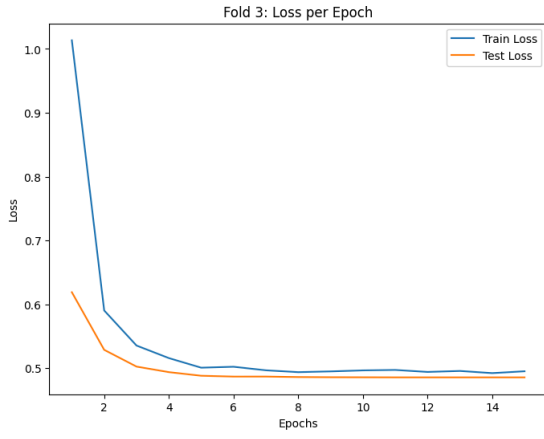
Şekil 37: Microsoft BeiT Epoch Loss Fold 1



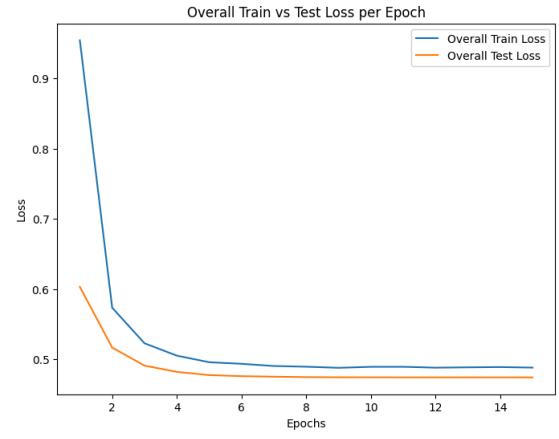
Şekil 38: Microsoft BeiT Epoch Loss Fold 2



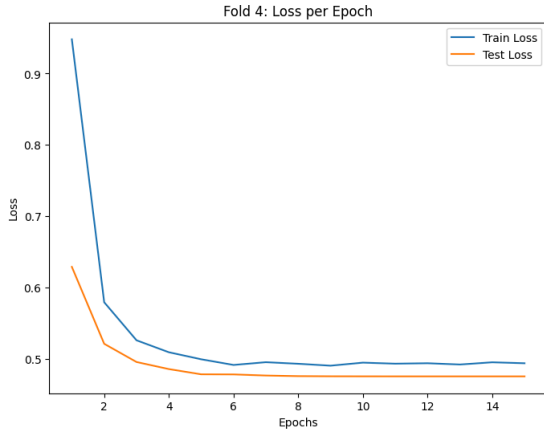
Şekil 41: Microsoft BeiT Epoch Loss Fold 5



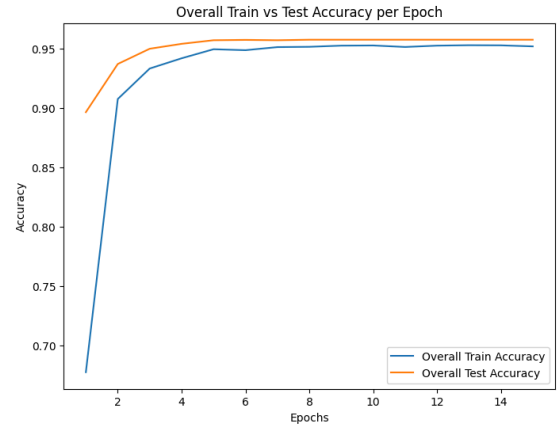
Şekil 39: Microsoft BeiT Epoch Loss Fold 3



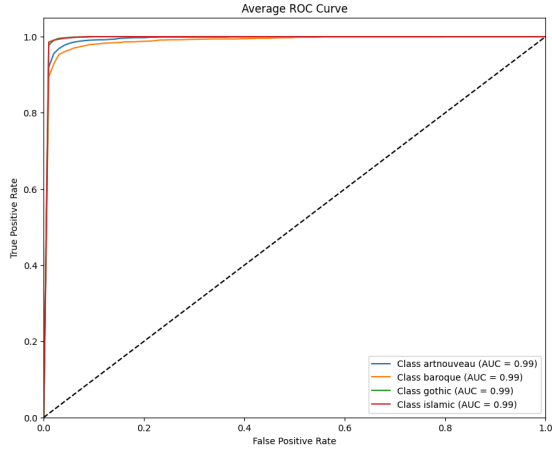
Şekil 42: Microsoft BeiT Overall Epoch Loss Grafiği



Şekil 40: Microsoft BeiT Epoch Loss Fold 4



Şekil 43: Microsoft BeiT Accuracy Grafiği

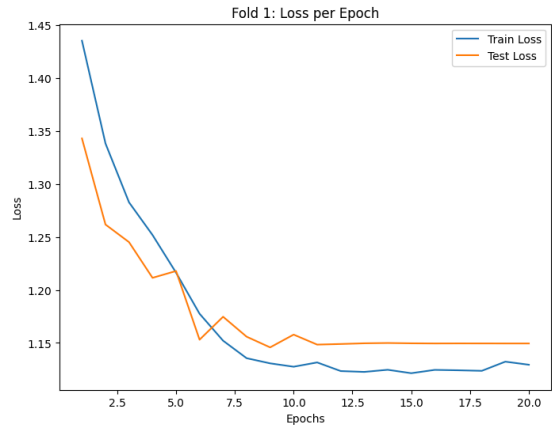


Şekil 44: Microsoft BeiT ROC Grafiği

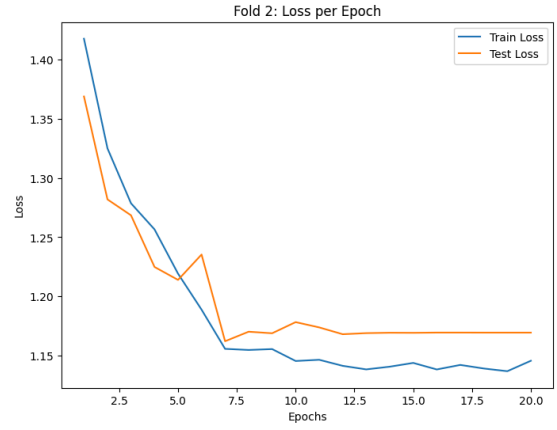
	ArtNouveau	Baroque	Gothic	Islamic	Overall
Precision	0.968100	0.911220	0.969920	0.983900	0.958285
Recall	0.919360	0.956920	0.978460	0.975140	0.957470
F1-score	0.943080	0.933460	0.974140	0.979440	0.957530
Accuracy	0.972220	0.965900	0.987000	0.989780	0.978725
Specificity	0.989860	0.968860	0.989900	0.994660	0.985820
MCC	0.925260	0.911060	0.965520	0.972700	0.943635
AUC	0.995580	0.990740	0.998860	0.999120	0.996075

Şekil 45: Microsoft BeiT Sonuç Metrikleri

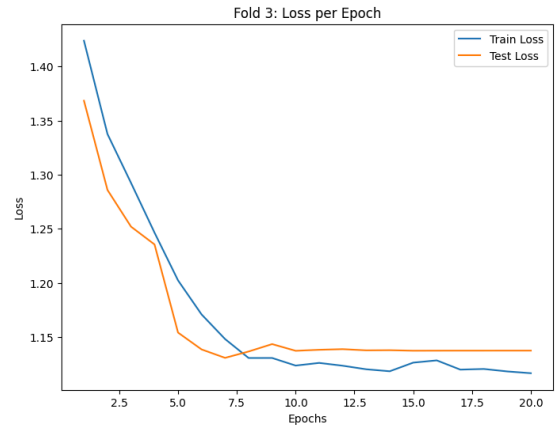
Bu modelden düzgün sonuç alabilmek için optimizasyon değerleriyle ilgili çok fazla deneme yapılması gerekmiştir, pek çok değere iyi yanıt vermeyen bir modeldi, kötü çıktılardan örnek bir çıktı aşağıda verilmiştir. Bu çıktıda dropout = 0.4 learning rate = 0.0006, weight decay=0.02 batch size = 8 verilmiştir. Sonuçlar arasındaki fark çok barizdir. Sadece bu değerlerle deneme yanılma yapılarak bu halden önceki verilen iyi hale getirilebilmiştir. Accuracy'ler görüldüğü üzere yüzde 50 civarlarında kalmıştır ve foldlardaki grafikler çok inişli çıkışlı seyretmektedir.



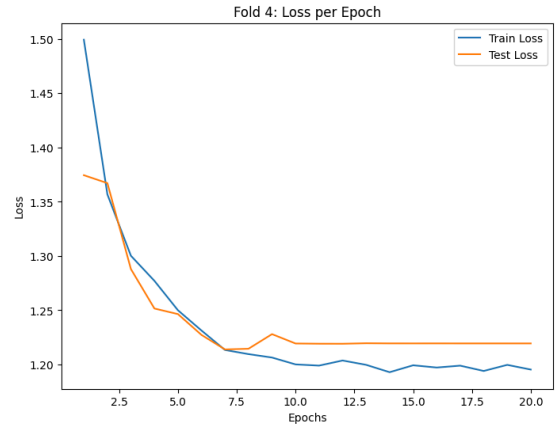
Şekil 46: Microsoft BeiT Kötü Epoch Loss Fold 1



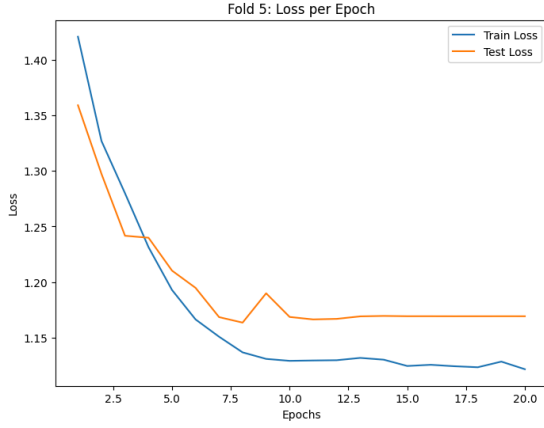
Şekil 47: Microsoft BeiT Kötü Epoch Loss Fold 2



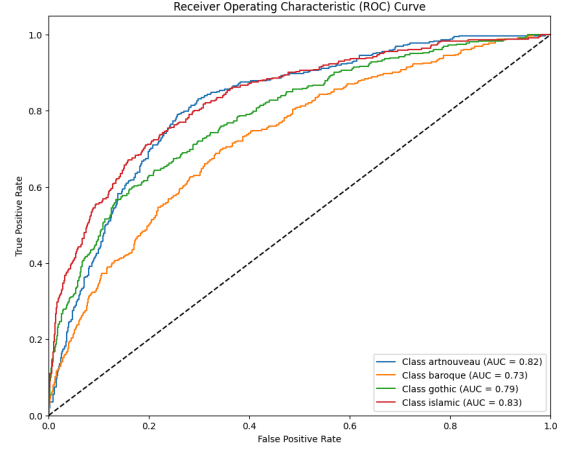
Şekil 48: Microsoft BeiT Kötü Epoch Loss Fold 3



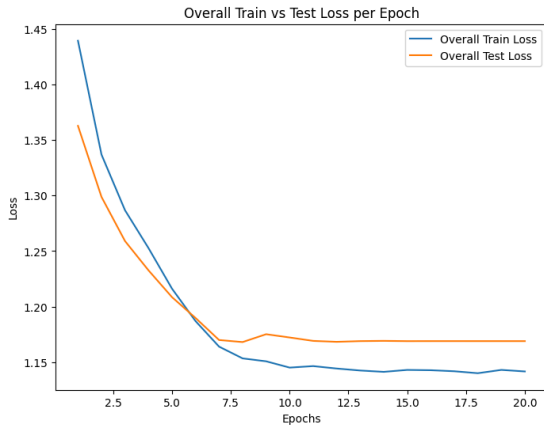
Şekil 49: Microsoft BeiT Kötü Epoch Loss Fold 4



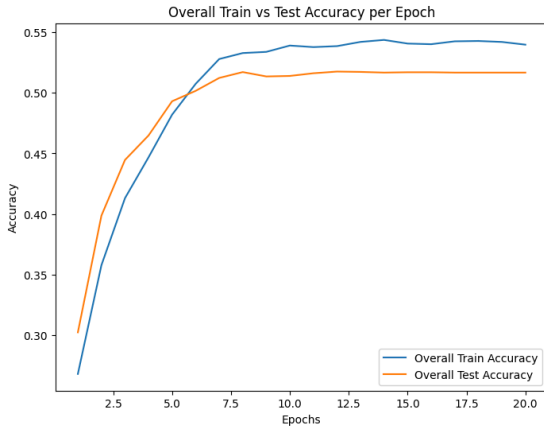
Şekil 50: Microsoft BeiT Kötü Epoch Loss Fold 5



Şekil 51: Microsoft BeiT Kötü ROC Grafiği



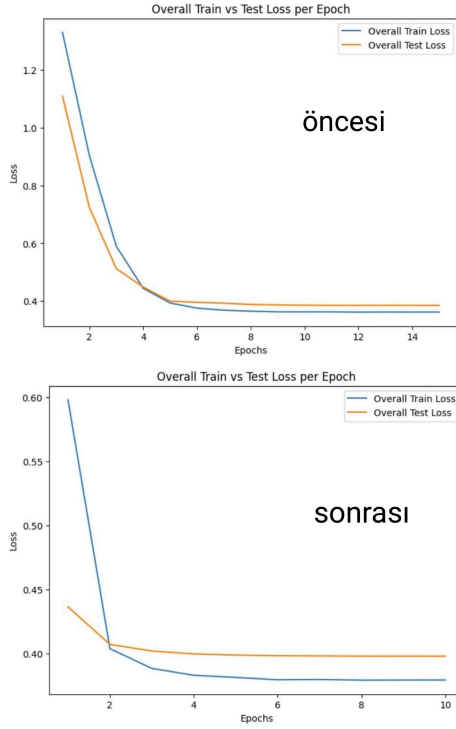
Şekil 51: Microsoft BeiT Overall Kötü Epoch Loss



Şekil 51: Microsoft BeiT Kötü Accuracy Grafiği

IV. KARŞILAŞILAN PROBLEMLER

Sonuçlar incelendiğinde lossların olması gerekenden daha yüksek kaldığı tespit edilmiştir, modelin lokal minimuma takıldığı düşünülmektedir. Dolayısıyla bu durumu iyileştirebilmek ve sonuçları global minimuma çekebilmek için bazı yöntemler denenmiştir,, bunlar farklı optimizasyon algoritmaları denemek, ve bu yeni algoritmalara momentum değerleri vermek olmuştur. Projede normalde AdamW optimizasyon algoritması kullanılmıştır, bu sorunu çözümlmek için SGD, RMSprop ve Adam algoritmaları da denenmiş, momentum değerleri değiştirilerek bunlar arasında geçişler yapılmıştır. Her ne kadar bazı denemelerde başlangıçtaki losslarda bir düşme gözlemlense de, genel sonuçlara bakıldığında çok bir iyileşme gerçekleştirilememiştir.



Şekil 52: Karşılaştırma

V. SONUÇ

Bu proje makine öğrenmesi ve derin öğrenme tekniklerini kullanarak farklı mimari akımların sınıflandırılmasını hedeflemiştir. Google'dan toplanan görsel veriler ile 5 farklı tranformer modeli kullanılarak eğitim yapılmış ve bu modellerin performansı değerlendirilmiştir. Sonuçlar göstermiştir ki:

- Google ViT, Microsoft Swin, Facebook DeiT, Microsoft BEiT ve Facebook LeViT modelleri, mimari akımları sınıflandırmada başarılı olmuştur.

-Eğitim sürecinde kullanılan dropout, learning rate ve weight decay gibi hiperparametrelerin optimize edilmesi modellerin genel performansını artırmıştır.

-K-fold cross-validation yöntemi, modelin genelleme yeteneğini değerlendirmede etkili olmuştur.

Sonuç olarak modellerinin mimari akımları tespit etme konusunda başarılı olduğu görülmüştür. İleride yapılacak çalışmalarda daha büyük ve çeşitlendirilmiş veri setleri kullanarak modellerin performansının daha da artırılabilceği düşünülmektedir. Ayrıca farklı optimizasyon teknikleri ve veri çoğaltma stratejileri kullanılarak modellerin yeteneklerinin iyileştirilebileceği öngörülmektedir.

REFERANSLAR

[1] Hugging Face Documentation, [Çevrimiçi]. Mevcut: <https://huggingface.co/docs>[Erişim: Haziran 05, 2024].

[2] Hugging Face Transformers Documentation, [Çevrimiçi]. Mevcut: <https://huggingface.co/transformers/>. [Erişim: Haziran 05, 2024].

[3] PyTorch Documentation, [Çevrimiçi]. Mevcut: <https://pytorch.org/docs/>. [Erişim: Haziran 05, 2024].

[4] Matplotlib Developers, [Çevrimiçi]. Mevcut: <https://matplotlib.org/stable/contents.html>. [Erişim: Haziran 05, 2024].

[5] Selenium Documentation, [Çevrimiçi]. Mevcut: <https://www.selenium.dev/documentation/>. [Erişim: Haziran 05, 2024].