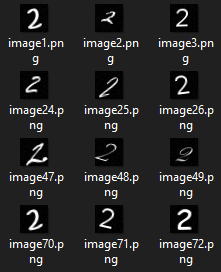
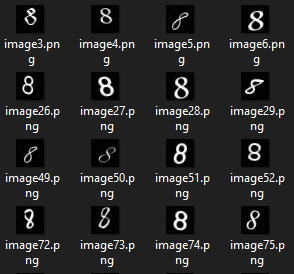
**Musa Akyüz**

**Bolu Abant İzzet Baysal Üniversitesi**

**Veri Madenciliği Proje Ödevi**

**1 – Veri seti bulma ve arff dosyası oluşturma aşaması.**

Matlab programında sınıflandırma için kullanılan ve hazır olarak bulunan, 28 x 28 x 1 boyutlarında 0’dan 9’a kadar olan sayıların farklı biçimlerde yazılmış, çizilmiş hallerinden 1000’er resim dosyası Veri seti zaten hali hazırda sınıflandırılmış şekilde klasörleştirilmiş. Bir python scripti ile rastgele seçilmiş resimlerden, Weka programının tanıyacağı arff formatında metin dosyası oluşturulacaktır. tekrar karıştırılarak



Text

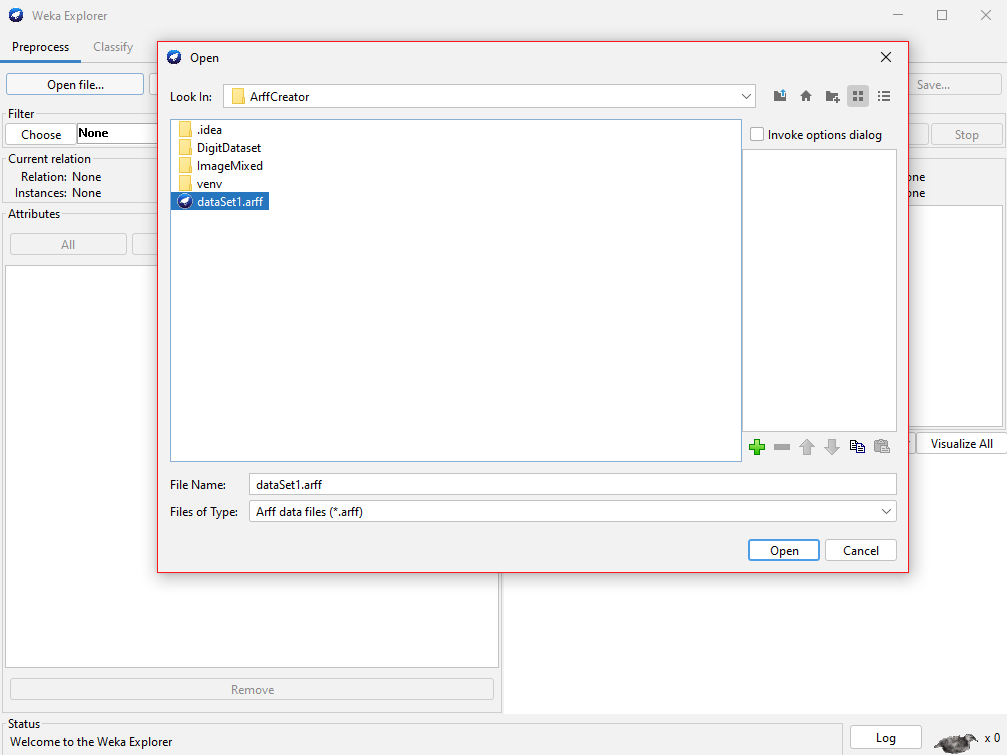
Description automatically generated

Text

Description automatically generated

Şekil – Resim 1 Örnek Veriler - Resim 2 Oluşturulan Arff dosyası - Resim 3 Arff dosyası oluşturan python kodu

Bu python kodu sayesinde 10.000 adet resmin dosya konumlarını .arff uzantılı dosya oluştururldu. Daha sonra bu .arff dosyasını (bkz Şekil 2) Weka programını kullanarak %30 test %70 eğitim olarak yani 7000 adet karışık resim dosyasını eğitimi verisi olarak, 3000 adet eğitimi dosyasını ise test seti olarak ayrılacaktır. Weka’da bu işlemi yaparken python kodu ile oluşturulan .arff dosyasını filtrelerden geçirdikten sonra veri setini bölümleme işlemi yapılmıştır.



Şekil Weka programından oluşturulan veri setinin seçilmesi

**2 – Anlamlı problem bulma aşaması**

Problem, bu veri setini kullanarak sayıları tanımasını sağlayarak sınıflandırmayı yapmak.

**3 – Algoritmalar, Kurallar, Yapılan İşlemler**

Weka programında resimler üzerinden eğitimi yapılırken harici resim filtreleme modülü indirilmeli.

Weka > Tools > Packet Manager > imageFilter yolundan resim filtrelerine ulaşılır. Bu filtreler resim üzerindeki R, G, B değerleri üzerinden birden çok sınıf oluşturur. Örneğin bu veri setini EdgeHistogram adında filtreye sokalım. Bu filtre resim üzerindeki kenarların, keskin geçişlerin açısına, uzunluğuna, eğikliğine gibi farklı bilgilere bakarak bir sonuç oluşturur (Bkz Şekil 3). Farkedileceği üzere bazı sınıflar sonuç olarak bir farklılık ortaya katmamış. Tüm veriler tek bir değer üzerinde toplanmış. Bu durum veri setimiz için uygunluğunun az olduğunu gösteriyor olabilir. Bu filtre ile ZeroR veya OneR gibi basit algoritmalar kullanıldığında sonucun (Bkz Şekil 4) felaket oranlarında kötü olduğunu görürüz. Ama J48 algoritması %93 oranda doğru sonuç veriyor.



Şekil : Edge Histogram filtrelemesi sonucu oluşan sınıfların bir kısmı

Table

Description automatically generated with medium confidenceText

Description automatically generated



OneR

ZeroR

Şekil 4 : OneR ve ZeroR algoritmalarının EdgeHistogram karşısında sonuçları

Bu kısımda imageFilterdaki filtreleri ile ZeroR, OneR, NaiveBayes ve J48 algoritmalarını kıyaslayalım. Hangi algoritma hangi filtre ile daha düzgün sonuçlar veriyor, inceleyeceğiz. Genel olarak tüm filtrelerde güzel sonuçlar çıkaran tek J48 karar yapısı var. Özellikle de Resample işlemi yapıldıktan sonra çok iyi sonuçlara ulaştı. ZeroR algoritması en çok etki eden tek bir sınıfı incelediği için her seferinde sadece bir değerin tam doğrusunu bulabiliyor. OneR ise veri setindeki diğer sınıfları da hesaba katarak işlemler yaptığından ZeroR a göre nispeten daha iyi.

(Bkz Tablo 1)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Karşılaştırma** | **ZeroR** | | **OneR** | | **Naïve Bayes** | | **J48** | |
| Positive | Negative | Positive | Negative | Positive | Negative | Positive | Negative |
| **Edge Histogram** | 10 | 90 | 20.3 | 79.7 | 71.4 | 28.6 | 93 | 7 |
| **Color Layout** | 10 | 90 | 24.4 | 75.6 | 58.2 | 41.8 | 97.6 | 2.4 |
| **Simple Color Histogram** | 10 | 90 | 24.5 | 75.5 | 23.2 | 76.8 | 50.8 | 49.2 |
| **Gobor Filter** | 10 | 90 | 26.7 | 73.3 | 15.7 | 84.3 | 66.3 | 33.7 |
| **Auto Color Correlogram** | 10 | 90 | 18.8 | 81.2 | 0 | 0 | 71.5 | 28.5 |
| **Jpeg Coefficient** | 10 | 90 | 21 | 79 | 28.7 | 71.3 | 89 | 11 |
| **Birleşim** | 10 | 90 | 24.4 | 75.6 | 77.8 | 22.2 | 96.5 | 3.5 |
| **Birleşim ve ReSample uygulandıktan sonra** | 10.7 | 89.3 | 24.9 | 75.1 | 79.1 | 20.9 | 98.7 | 1.3 |

Tablo 1 : Çeşitli testler

Bu tobloyu oluşturduktan sonra, bileşik filtreye resample uygulanmış olan veri seti kullanılması tercih edilir. Bu veri setini hazırlamak için yine Weka programındaki veri seti bölüştürme filtresi kullanılır.

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, application

Description automatically generated



Şekil 5: Eğitim ve Test veri setleri bölüştürme işlemi

**4 – Eğitim, Test Sonuçları**

A picture containing table

Description automatically generated



Şekil 6: Eğitim sonucu

Table

Description automatically generated



Şekil 7: Test sonuçları