Her bir görselin CSV’ye çevrilmesi ve metriklerin belirlenmesi, genellikle aşağıdaki genel adımlarla yapılır. Bu süreçte görsellerden belirli özellikler çıkarılır ve bu özellikler sınıflandırma modeli için uygun hale getirilir.

**1. Görsellerin Yüklenmesi**

* Görseller, belirli bir dizinden yüklenir.
* Yüklemek için Python’da **OpenCV**, **Pillow (PIL)** veya **Scikit-Image** gibi kütüphaneler kullanılır.

**2. Özellik Çıkarma (Feature Extraction)**

Görsellerden özellikler çıkarılır. Bu özellikler görselleri sınıflandırmada önemli ipuçları sağlar.

**a. Temel İstatistiksel Özellikler**

* **Mean (Ortalama):** Görseldeki piksellerin ortalama değeri.

Python:

mean = image.mean()

* **Variance (Varyans):** Piksel değerlerinin değişkenliği.

Python:

variance = image.var()

* **Entropy (Entropi):** Görselin bilgi içeriği. Daha karmaşık görseller daha yüksek entropiye sahip olur.

Python:

from skimage.measure import shannon\_entropy

entropy = shannon\_entropy(image)

**b. Dokusal Özellikler (Texture Features)**

* **GLCM (Gray-Level Co-occurrence Matrix):** Görüntüdeki dokusal ilişkileri analiz eder.
  + **Contrast, Energy, Homogeneity, Correlation** gibi metrikler hesaplanabilir.

**c. Şekil ve Geometri Özellikleri**

* Görüntüdeki tümörün boyutunu ve şeklini analiz etmek için bölgesel özellikler kullanılır.
  + **Alan (Area)**, **Çevre (Perimeter)**, **Yuvarlaklık (Circularity)** gibi metrikler.

**d. Dönüşüm Tabanlı Özellikler**

* Fourier veya Dalga Dönüşümü (Wavelet Transform) ile frekans bazlı özellikler çıkarılır.

**3. Verilerin Etiketlenmesi**

* Her görselin hangi sınıfa ait olduğu belirlenir (örneğin, tümör var/yok, benign/malign).
* Bu etiketler CSV’ye yazılır.

**4. CSV Oluşturma**

Elde edilen tüm metrikler bir CSV dosyasına yazılır.

**Metriklerin Belirlenme Kriterleri**

Metrikler, sınıflandırma probleminin doğasına göre seçilir:

* **Tümör Sınıflandırma**: Doku analizi (entropy, contrast).
* **Bölgesel Analiz**: Şekil ve boyut özellikleri (area, perimeter).
* **Görselin Genel Özellikleri**: Piksel tabanlı istatistiksel değerler (mean, variance).

CSV dosyasındaki bilgiler şunları gösteriyor:

1. **Veri Yapısı:**
   * CSV dosyasında **3762 satır** ve **15 sütun** bulunuyor.
   * Sütunlardan bazıları:
     + **Image**: Görsel adı veya yolu.
     + **Class**: Görselin sınıf etiketi (örneğin, 0 veya 1 gibi).
     + Görsellerden elde edilen istatistiksel özellikler (örneğin, Mean, Variance, Entropy vb.).
2. **Görsellerin Özellikleri:**
   * Görsellerden çeşitli metrikler hesaplanmış:
     + **Mean (Ortalama)**: Piksel değerlerinin ortalaması.
     + **Variance (Varyans)**: Piksel değerlerinin dağılım genişliği.
     + **Entropy (Entropi)**: Görselin bilgi içeriği.
     + **Contrast (Kontrast)** ve diğer metrikler.
   * Bu özellikler, görsellerin sınıflandırılmasında kullanılabilecek verilerdir.
3. **Veri Hazırlama Mantığı:**
   * Görseller öncelikle istatistiksel veya görüntü işleme yöntemleriyle sayısal değerlere dönüştürülmüş.
   * Örneğin, Python’da **OpenCV**, **Pillow**, veya **scikit-image** gibi kütüphaneler kullanılarak piksellerden bu özellikler hesaplanmış olabilir

Daha fazla sınıflandırma metriği (özellik veya **feature**) kullanmak her zaman daha iyi bir sınıflandırma performansı anlamına gelmez. Bunun etkisi şu faktörlere bağlıdır:

**1. Özelliklerin Kalitesi**

* **Özelliklerin anlamlı olması**, sınıflandırma performansı için çok önemlidir.
  + Faydalı ve ayırt edici özellikler sınıflar arasında net bir ayrım sağlar.
  + Fazla ancak **gürültü içeren** veya **ilgili olmayan** özellikler, modelin performansını düşürebilir (overfitting'e yol açabilir).
* Örnek:
  + Görüntü kontrastı (contrast) ve entropi genelde görsel analizi için anlamlıdır.
  + Ancak rastgele bir özellik eklenirse, sınıflandırmayı zorlaştırabilir.

**2. Özelliklerin Sayısı ve "Curse of Dimensionality"**

* Çok fazla özellik, yüksek boyutluluk problemini doğurabilir.
  + Yüksek boyutlarda, veri noktalarının arasındaki mesafeler homojenleşir, bu da sınıflandırma için zorluk yaratır.
  + Daha fazla özellik, daha fazla veri gerektirir. Veriniz azsa, model doğru öğrenemez.
* Bu yüzden **özellik seçimi** yapılır:
  + **PCA (Principal Component Analysis)** veya benzeri tekniklerle boyut indirgeme.
  + Yüksek korelasyona sahip olan ya da düşük varyans gösteren özellikleri eleme.

**3. Modelin Karmaşıklığı ve Overfitting**

* Çok fazla özellik, modelin karmaşıklığını artırabilir:
  + Karmaşık bir model, veriye aşırı uyum sağlayarak test verisinde kötü performans gösterebilir (overfitting).
  + Daha basit ve anlamlı özellikler genelde daha iyi genelleştirme sağlar.
* Örnek:
  + Basit modeller (Logistic Regression) daha az ve anlamlı özelliklerle daha iyi çalışabilir.
  + Karmaşık modeller (Deep Learning), özelliklerin anlamını kendi başına çıkarabilir.

**4. Özelliklerin İlgisi ve Çeşitliliği**

* Özelliklerin **birbirinden bağımsız** olması önemlidir. İki özellik aynı bilgiyi içeriyorsa, sadece biri yeterlidir.
* **Çeşitli metrikler** farklı bakış açıları sağlar ve modeli güçlendirebilir:
  + Örneğin, bir görselde hem piksel yoğunluğu (Mean) hem de kenar özellikleri (Edge Detection) kullanılırsa, daha iyi sonuç alınabilir.

**Sonuç: Kalite > Sayı**

* Özelliklerin sayısından çok **kalitesi** ve sınıflar arasındaki farkı ne kadar iyi yansıttıkları önemlidir.
* Fazladan özellikler eklemek yerine, anlamlı ve işlenebilir özellikler belirlemek daha iyi sonuç verir.
* Özellikleri seçerken ve değerlendirirken şu yöntemler faydalı olabilir:
  + Özellik seçimi algoritmaları (Forward Selection, Backward Elimination).
  + Özelliklerin model performansına etkisini ölçmek (örneğin, SHAP değerleri).