



2024-2025 Bahar Yarıyılı

Bilgisayar Mühendisliğinde Özel Konular Dersi-3

2. Ödevi

Öğrenci Adı-Soyadı: Mert Uz

Öğrenci Numarası: 22011036

Ders Yürütücüsü: Prof. Dr. Mine Elif Karslıgil

Yöntem

Ödevde CIFAR-10 database'i kullanılarak ve resimlerin renk benzerliklerini değerlendirerek bir resme en çok benzeyen 5 adet resmi tespit edilmesi istenmektedir. Bunun için **extract_cifar10()**, **load_cifar10_data()** ve **prepare_data()** fonksiyonları kullanılmıştır.

extract_cifar10()

Eğer CIFAR-10 veri klasörü mevcut değilse:

Sıkıştırılmış .tar.gz dosyası açılır

Tüm içeriği hedef dizine çıkartılır

load_cifar10_data()

Eğitim verileri için:

Her batch dosyası açılır ve içindeki veriler ve etiketler okunarak birleştirilir

Test verisi için:

test_batch dosyası açılır ve içindeki veriler ve etiketler okunur

Veriler:

Her görüntü 3072 boyutlu bir vektördür (1024 R + 1024 G + 1024 B)

RGB kanalları ayrılarak 32x32x3 görüntüler haline dönüştürülür

Etiketler: Her görüntünün sınıf numarasıdır

Veriler ve etiketler tuple olarak döndürülür

prepare_data()

Her görüntü için:

R, G, B kanalları ayrılır. Her kanal için histogram hesaplanır (256 olasılık değeri)

3 histogram birleştirilerek 768 boyutlu vektör elde edilir

Tüm görüntüler için bu işlem tekrarlanır

Sonuç: Tüm görüntüleri temsil eden histogram vektörleri listesi döndürülür

Ödevde kullanılacak olan resimler hazırlandıktan sonra eğitim için kullanılacak olan 250 resmin renk histogramı çıkarılıp normalize edilmelidir. Bu işlemler için **calculate_histogram(channel_data)** ve **normalized_histograms(images)** fonksiyonları kullanılmıştır.

calculate_histogram(channel_data)

Girdi: channel_data (tek bir renk kanalı, örn. R kanalı)

Boş bir histogram dizisi oluştur (256 elemanlı, her biri 0)

Tüm pikseller için:

Piksel değerini al (0–255 arası)

Bu değerin bulunduğu indeksi 1 arttır

Çıkış: 256 elemanlı histogram vektörü

normalized_histograms(images)

Girdi: images (32x32x3 boyutunda çok sayıda renkli görüntü)

Her bir görüntü için:

R, G ve B kanalları ayrı ayrı alınır

Her kanal için calculate_histogram() fonksiyonu çağrılır

3 kanalın histogramları birleştirilir → 768 boyutlu bir vektör elde edilir

Tüm görüntüler için bu işlem tekrarlanır

Çıkış: Tüm görüntülerin 768 boyutlu histogram vektörlerini içeren liste

Hem eğitim hem de test resimlerinin normalize edilmiş renk histogramları hesaplandıktan sonra, her test resmi için eğitim resimleri arasındaki euclidean distance'ı hesaplanmıştır. Bu bilgi ile euclidean distance'ın en az olan (en benzer) 5 resim, o test resminin en çok benzediği 5 resim olarak kullanıcıya çıktı olarak verilmiştir.

Euclidean distance hesaplamak için **euclidean_distance()** fonksiyonu, euclidean distance ile karşılaştırma yapıp en benzer 5 resmi bulmak için ise

find_top5_nearest_neighbors(test_hist, train_hists) fonksiyonu kullanılmıştır.

visualize_nearest_neighbors() fonksiyonuyla benzer ve asıl resim kullanıcıya çıktı oluşturmak için çağırılmıştır.

euclidean_distance()

Girdi: hist1, hist2 (iki ayrı 768 boyutlu histogram vektörü)

Fark vektörü = hist1 - hist2

Karekök(Toplam(fark vektöründeki elemanların karesi))

Çıkış: hist1 ve hist2 arasındaki Euclidean mesafesi (float değer)

find_top5_nearest_neighbors(test_hist, train_hists)

Girdi:

test_hist → test görüntüsünün histogram vektörü

train_hists → eğitim verilerine ait histogram vektörleri listesi

Boş bir liste oluştur: mesafe_listesi = []

Tüm eğitim histogramları için:

Euclidean distance hesaplama: $d = \text{euclidean_distance}(\text{test_hist}, \text{train_hist})$

(d, index) şeklinde listeye ekle

mesafe_listesini mesafeye göre küçükten büyüğe sırala. İlk 5 elemanın indekslerini al → en yakın 5 komşu

Çıkış: En yakın 5 komşunun indeksleri listesi

visualize_nearest_neighbors()

Girdi:

test_image → test için kullanılan görüntü

train_images → eğitim verisindeki tüm görüntüler

top5_indices → en benzer 5 görüntünün indeksleri

top5_indices içindeki her index için:

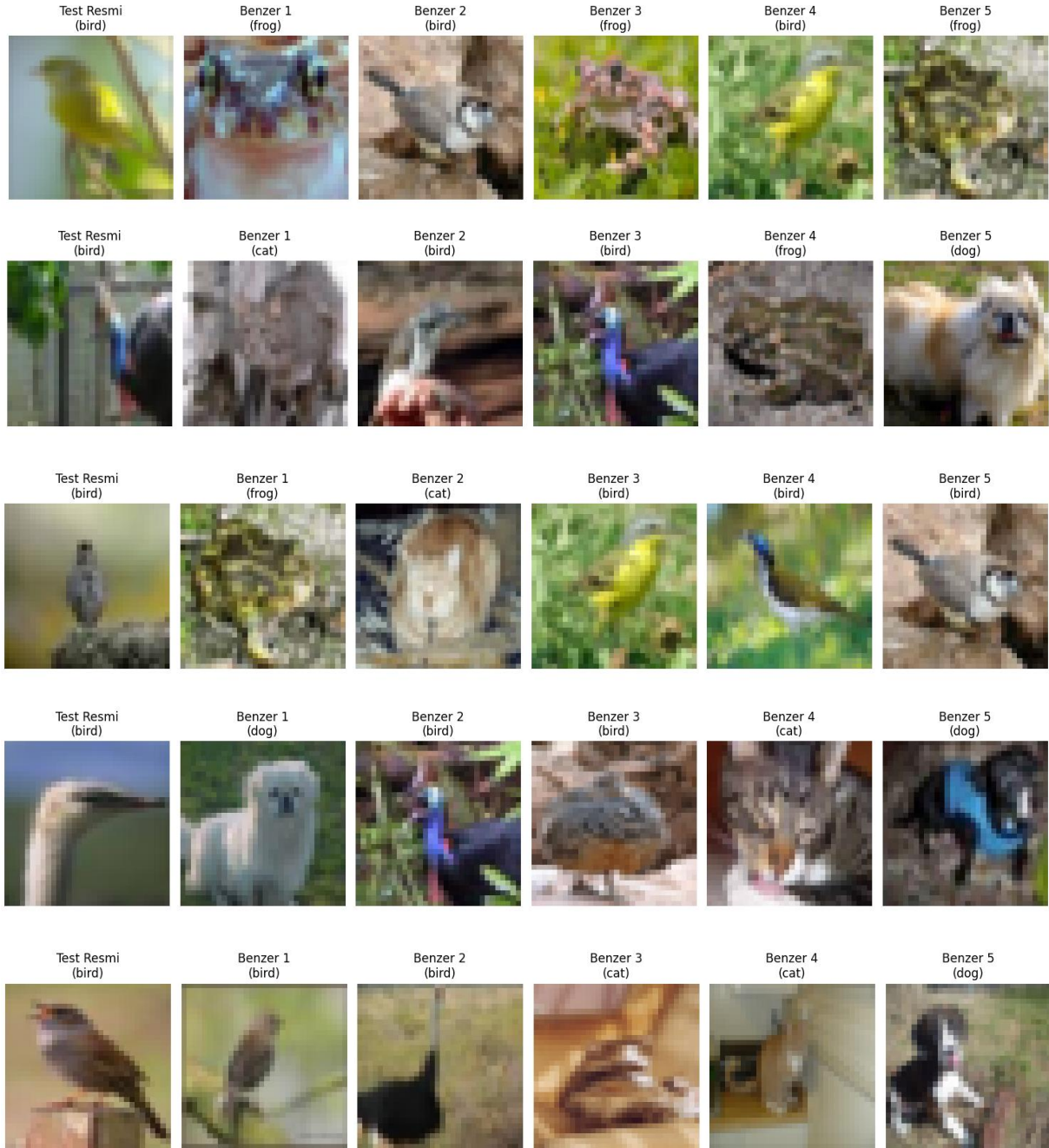
Karşılık gelen train_images[index] görüntüsünü sırayla ekle

Başlıklarla birlikte görselleştirmeyi göster

Çıkış: Grafik arayüzünde görsel olarak benzerlik çıktısı

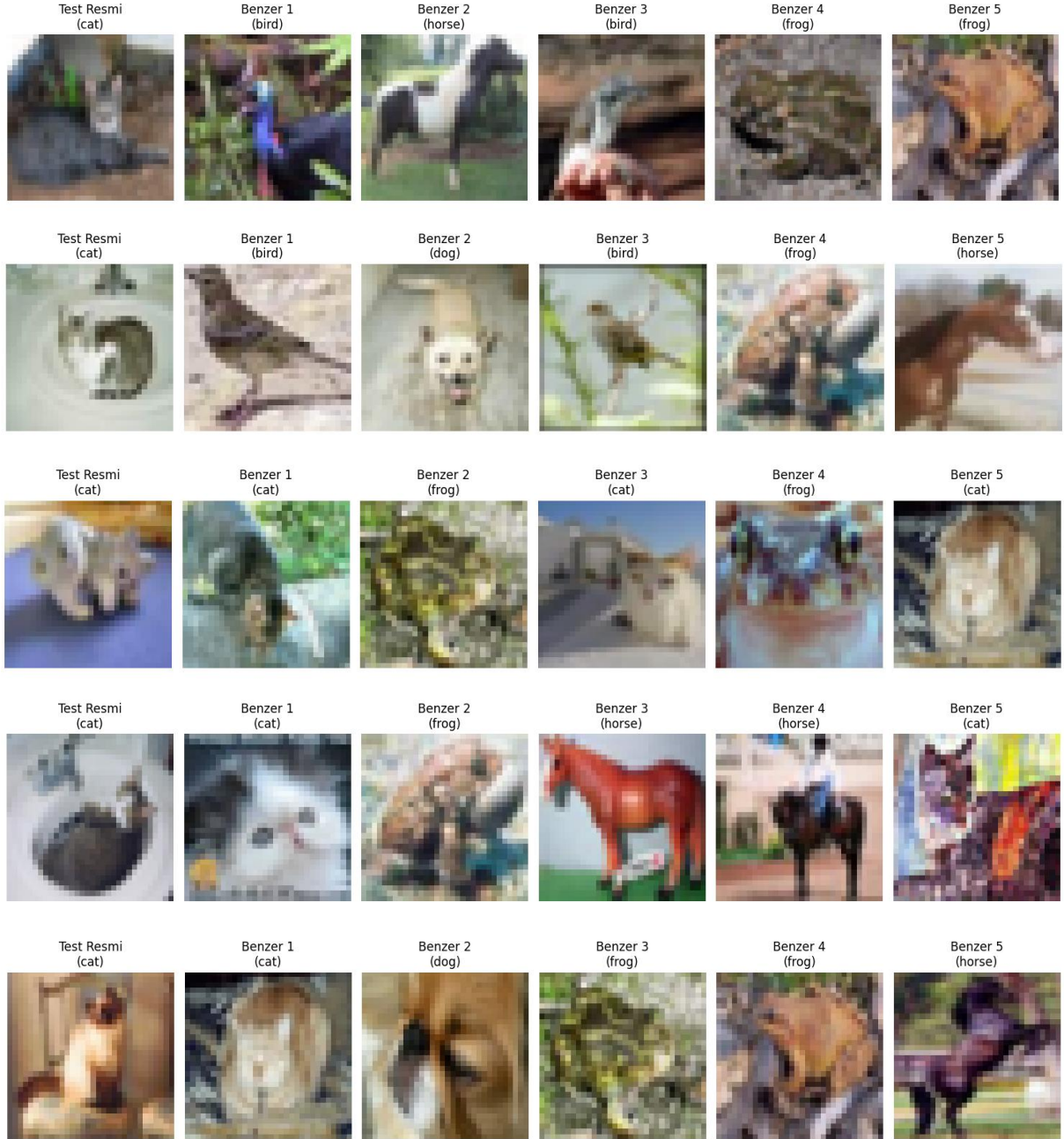
Uygulama

Bird sınıfı çıktıları



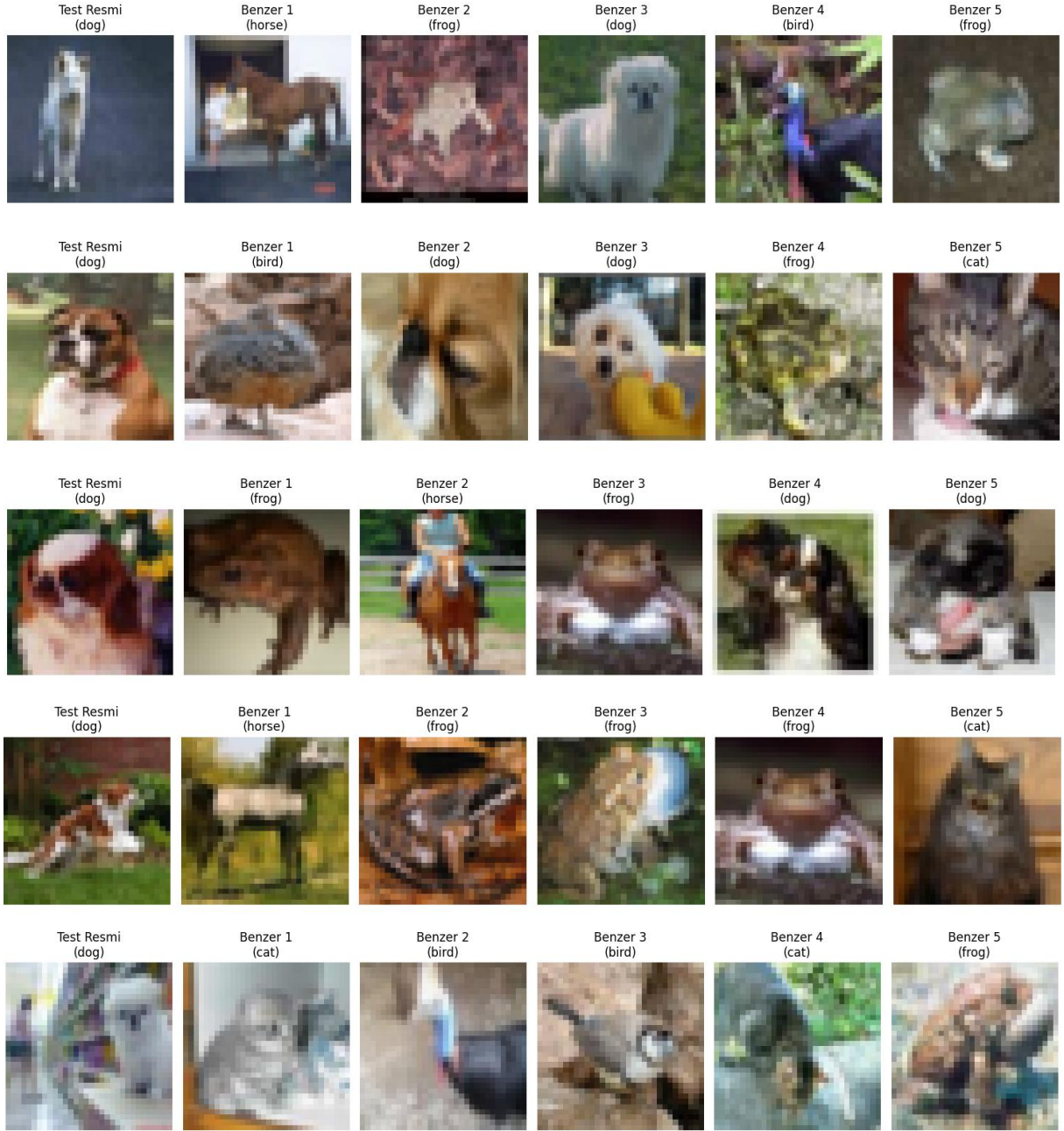
Eğer 5 resimden en az 1'i test resmi ile aynı sınıfta ise doğru sonuç, hiç benzer yok ise yanlış sonuç olarak hesaplanması istenmiş. Buna göre bird sınıfı için başarı oranı 5/5 yani %100'dür.

Cat sınıfı çıktıları



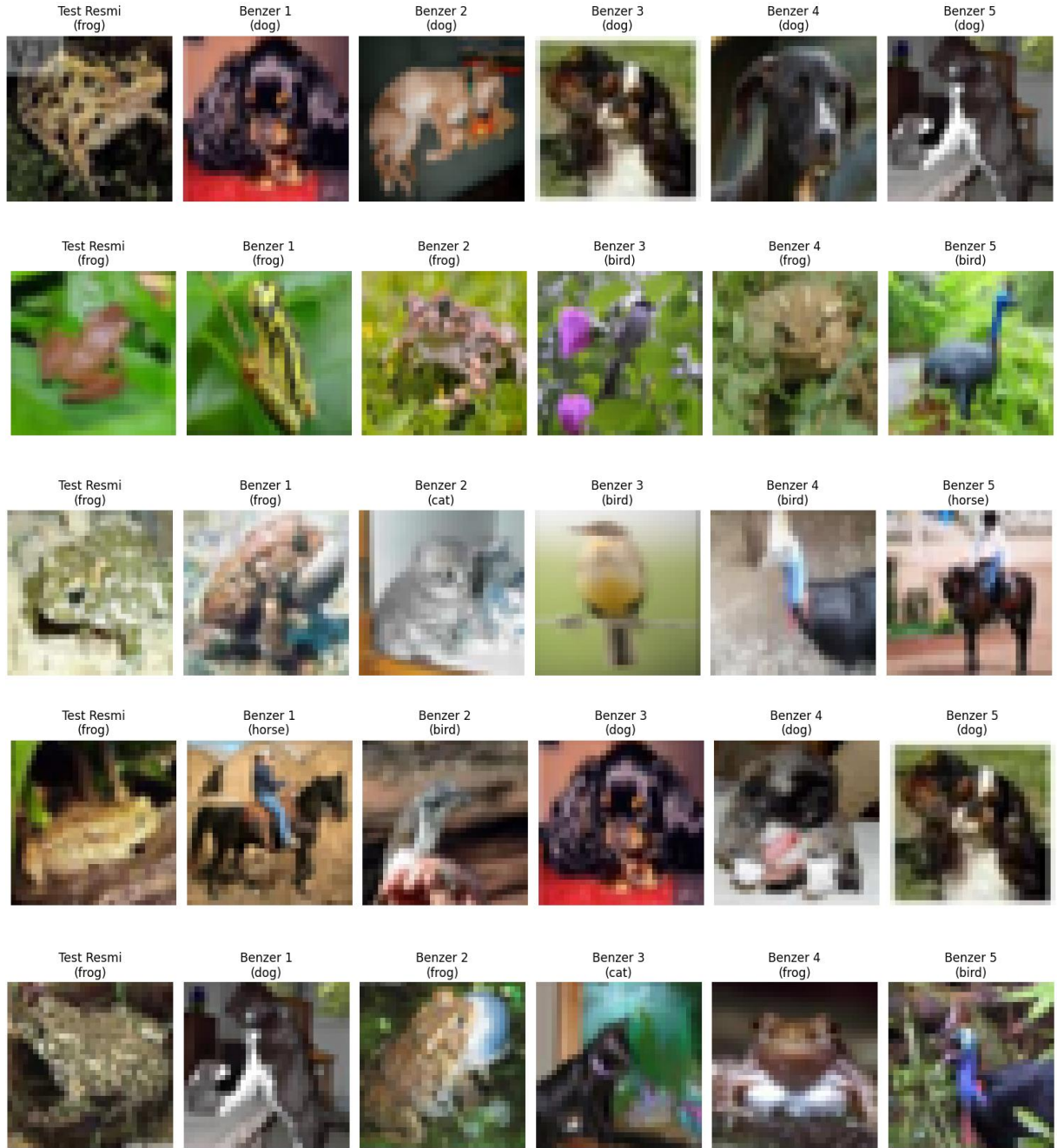
Cat sınıfı için başarı oranı 3/5 yani %60'tır.

Dog sınıfı çıktıları



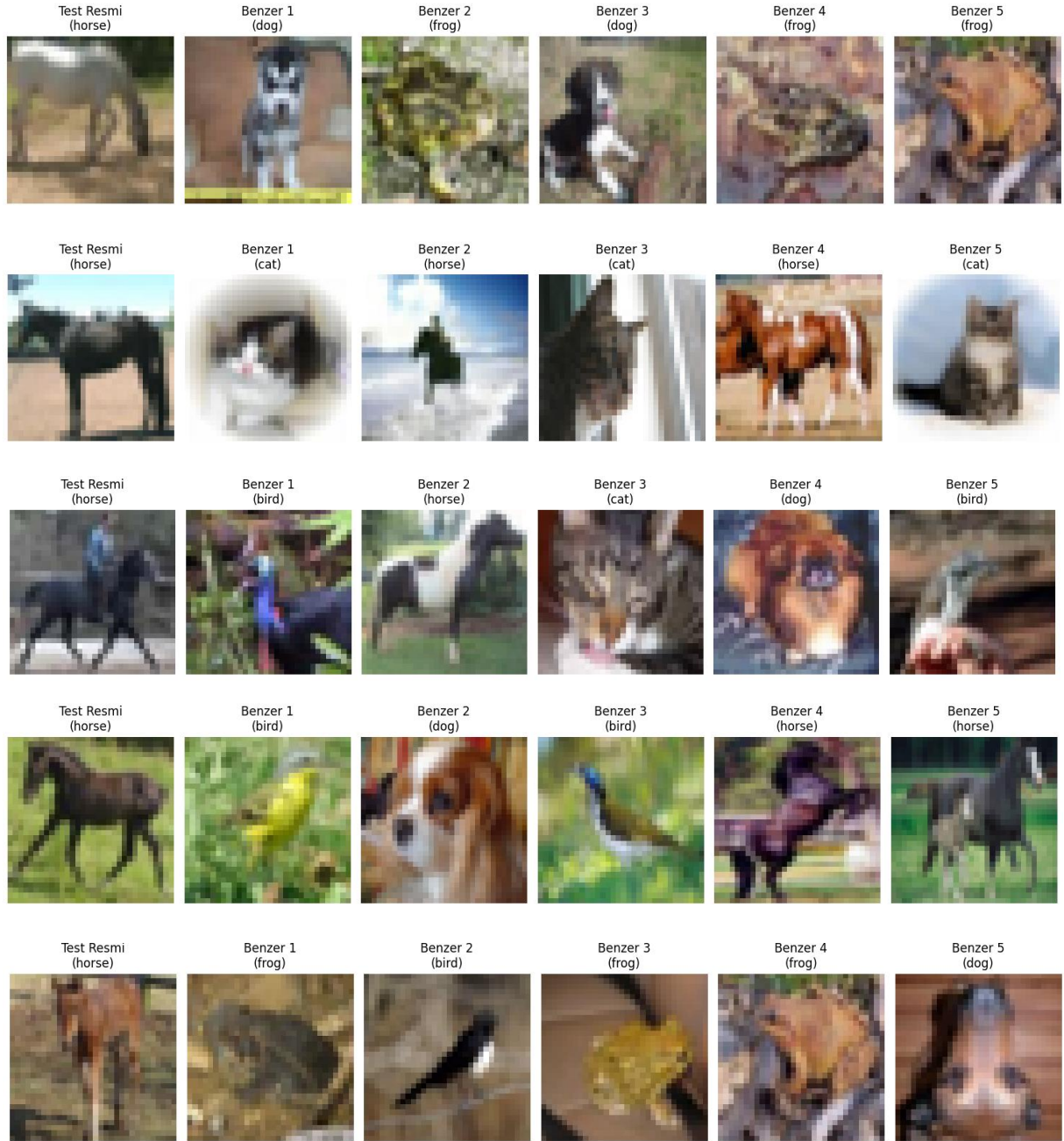
Dog sınıfı için başarı oranı 3/5 yani %60'tır.

Frog sınıfı çıktıları



Frog sınıfı için başarı oranı 3/5 yani %60'tır.

Horse sınıfı çıktıları



Horse sınıfı için başarı oranı 3/5 yani %60'tır.

Toplam başarı yüzdesi: 17/25 yani %68'dir.

Sonuç

Renk tabanlı histogram yöntemi, özellikle arka planları veya genel renk tonları benzer olan sınıflarda başarılı sonuçlar vermiştir. Örneğin deniz, gökyüzü veya bitki örtüsü gibi belirgin renk desenlerine sahip sınıflarda doğru eşleşmeler gözlemlenmiştir. Ancak, nesne şekli ve yapısal detayların belirleyici olduğu sınıflarda renk benzerliği yeterli olmamış ve bazı yanlış eşleşmeler oluşmuştur.

Avantajları:

- Basit ve hızlı bir yöntemdir.
- Görüntülerin yapısal analizi yapılmadan da temel benzerlik ölçümü yapılabilir.
- Hafızada az yer kaplayan öznitelikler (768 boyutlu vektör) ile çalışır.

Dezavantajları:

- Nesne şekli ve yapısal bilgi dikkate alınmadığı için görsel olarak farklı ama renk olarak benzer olan nesneler yanlış eşleştirilebilir.
- Arka planın baskın olduğu görüntülerde, arka plan rengi benzerliği yanıltıcı olabilir.
- Işıklandırma farklılıkları histogramları etkileyebilir.

Yöntem, temel seviye bir görsel benzerlik tespiti için yeterlidir; ancak daha doğru ve anlamlı sonuçlar için **şekil bilgisi**, **kenar tespiti**, **özellik çıkarımı** gibi daha gelişmiş tekniklerin de entegre edilmesi gerekmektedir. Bu çalışma, renk tabanlı özelliklerin belirli durumlarda faydalı olabileceğini, ancak tek başına yetersiz kalabileceğini göstermiştir.