Indian Journal of Animal Sciences **93** (4): 401–405, April 2023/Article
https://doi.org/10.56093/ijans.v93i04.131010

Principal component analysis in pig breeds identification

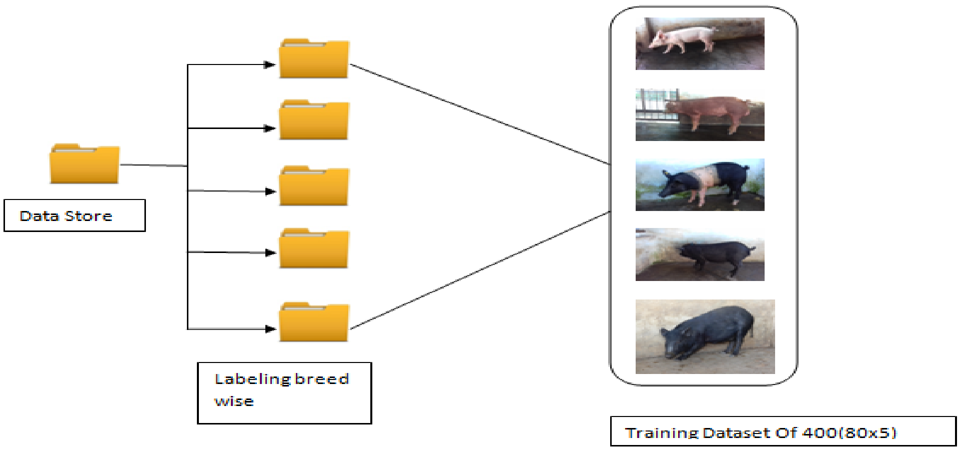
SANKET DAN[1✉], SATYENDRA NATH MANDAL[1], PRITAM GHOSH[1], SUBHRANIL MUSTAFI[1] and SANTANU BANIK[2]

Kalyani Government Engineering College, Kalyani, Nadia, West Bengal 741 235 India

Received: 2 December 2022; Accepted: 24 March 2023

Supplementary Fig. 1. Capturing Ghungroo pig in uncontrolled environment.

Supplementary Fig. 2. Capturing Yorkshire pig in controlled environment.



Supplementary Fig. 3. Data Labelling.

Supplementary Fig. 4. Pig breeds used in this research.

Input: RGB images from five pig breeds with size = (300 X 300) px
Result: Template (Central mean (ψ), Image space (B_j) and Projected train image vector ($project_{train\_img}[]$)

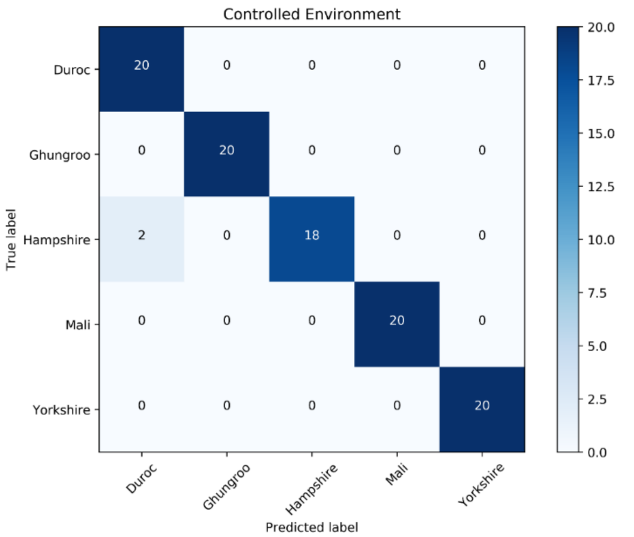
/* n = total number of images */
/* Images[] = image datastore */
for $i = 1 : n$ **do**
  /* Convert RGB image into grey scale image */
  $img = rgb2gray(Images[i])$
  /* P = row, Q = column */
  $(P, Q) = sizeof(img)$
  /* convert each 2D image into 1D column vector */
  $temp = reshape(img^t, (P * Q), 1)$
  $X[i] = temp$
end
/* Compute the central mean vector ψ */
$\psi = \sum_{n=1}^{P} X$
/* mean vector ψ is subtracted from X */
$A[i] = X[i] - \psi$
/* Compute co-variance matrix */
$C = A^t * A$
/* Calculate eigen vectors(V) and eigen values(D) of C */
$[V, D] = eigen(C)$
$m = count(D)$
/* Dropping columns in V for D <= 1 */
for $i = 1 : m$ **do**
  if $D[i, i] > 1$ **then**
    $L_{eigen}[i] = V[i]$
  end
end
/* Compute image space B_j by multiplying X with L_{eigen} */
$B_j = X * L_{eigen}$
/* Project X into B_j */
for $i = 1 : m$ **do**
  $\omega = B_j^t * X[i]$
  $project_{train\_img}[i] = \omega[i]$
end
return $\psi, B_j, project_{train\_img}[]$

Supplementary Fig. 5. Development of pig breeds prediction model.

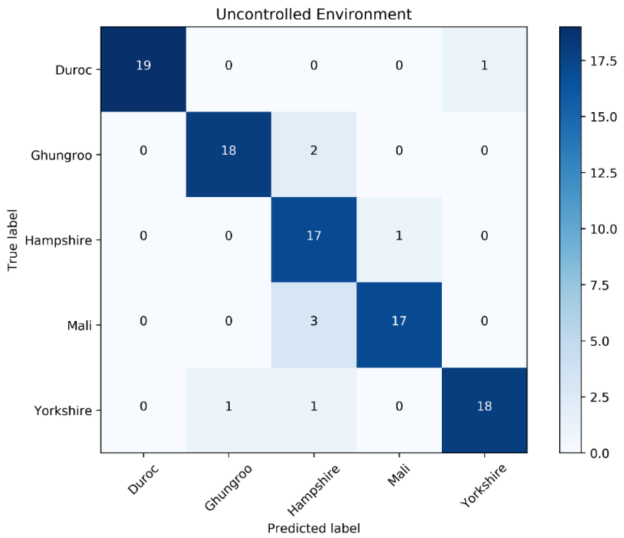
Input: Any RGB image from test set with size = (300 X 300) px, Central mean (ψ), Image space (B_j), Projected train image vector ($project_{train\_img}[]$) and Image datastore (Images[])
Result: Breed of test image (pigbreed)

/* img = test image */
/* Convert RGB image into grey scale image */
$img = rgb2gray(img)$
/* P = row, Q = column */
$(P, Q) = sizeof(img)$
/* convert 2D image into 1D column vector */
$X_{test} = reshape(img^t, (P * Q), 1)$
/* substract ψ from X_{test} */
$X_{test} = X_{test} - \psi$
/* compute projected test image vector p_{test_img} */
$p_{test\_img} = B_j^t * X_{test}$
/* compute Euclidean distance (e) from $project_{train\_img}[]$ and p_{test_img} */
/* len = number of columns in B_j */
$e = []$
for $i = 1 : len$ **do**
  $temp = compute\_euclidean(project_{train\_img}[], p_{test\_img})$
  e[i] = temp
end
/* Find minimum Euclidean distance (e_{min}) from e and store the corresponding index in id_{index} */
$(e_{min}, id_{index}) = minimum(e)$
/* Find the label of the image (pigbreed) corresponding to id_{index} from Images[] */
$pigbreed = label(Images[id_{index}])$
return pigbreed

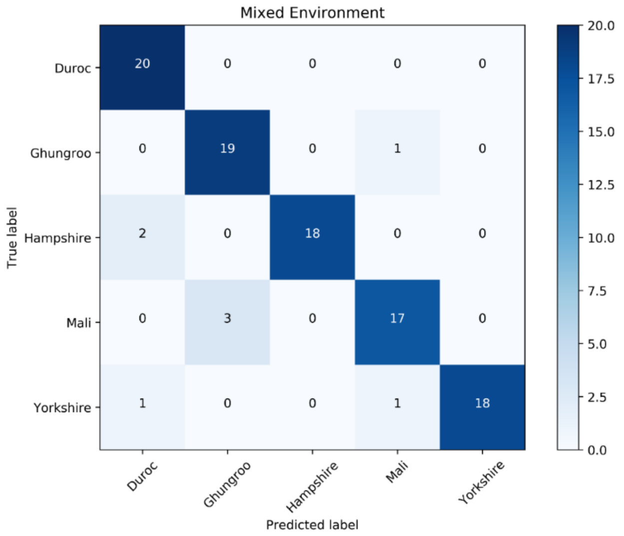
Supplementary Fig. 6. Breed prediction.



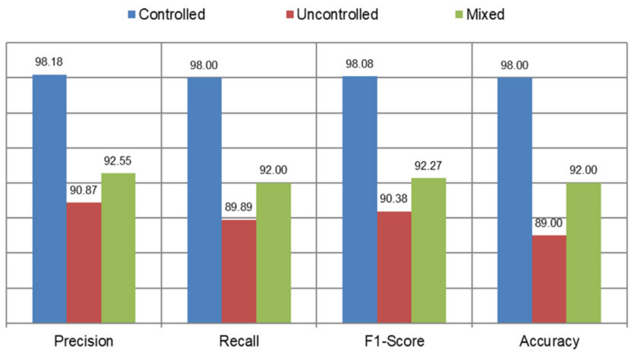
Supplementary Fig. 7. Confusion matrix for controlled environment.



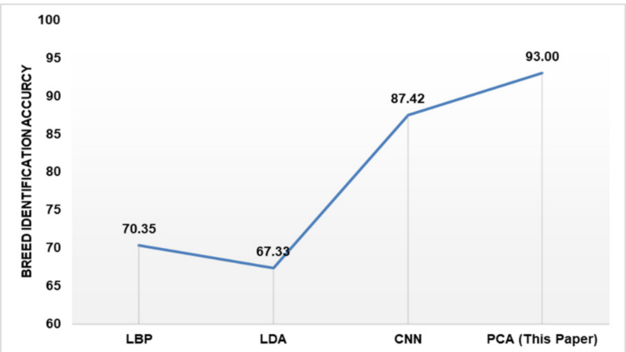
Supplementary Fig. 8. Confusion matrix for uncontrolled environment.



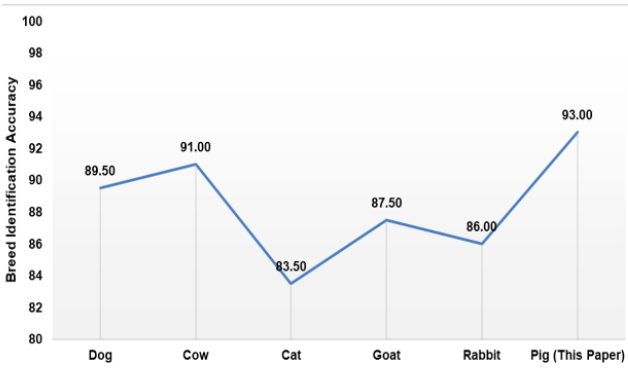
Supplementary Fig. 9. Confusion matrix for mixed environment.



Supplementary Fig. 10. Graphical representation of comparisons among controlled, uncontrolled and mixed sets.



Supplementary Fig. 11. Graphical representation of comparison among different breed prediction algorithms.



Supplementary Fig. 12. Graphical representation of comparison of different PCA-based algorithms used in breed prediction.