APLICAÇÃO DO MÉTODO DE ANÁLISE DE COMPONENTES PRINCIPAIS - PCA

Temática: Análise Comparativa Entre o Uso de Bandas Espectrais e o Uso da Análise de Componentes Principais (ACP) na Classificação de Uso e Cobertura da Terra

O objetivo deste estudo de caso é comparar e avaliar a potencialidade da técnica de Análise de Componentes Principais para o aprimoramento da acurácia da classificação do uso e cobertura do solo, em uma área de estudo situada no sertão da Paraíba, totalizando uma área de 9730,00Km^2.

Figura 01: Localização da área de estudo

Para esta análise, foi adquirida imagem orbital do satélite Landsat-8, através do Serviço Geológico dos Estados Unidos (USGS), com data de imageamento em 29/11/2023 e com o instrumento imageador Operational Land Imager (OLI), nas bandas espectrais B2 (Faixa espectral do Visível Azul, 0.45 - 0.51 µm), B3 (Faixa espectral do Visível Verde, 0.53 - 0.59 µm), B4 (Faixa espectral do Visível Vermelho, 0.64 - 0.67 µm) e B5 (Faixa espectral do Infravermelho próximo, 0.85 - 0.88 µm), todas as bandas com resolução espacial de 30 m.

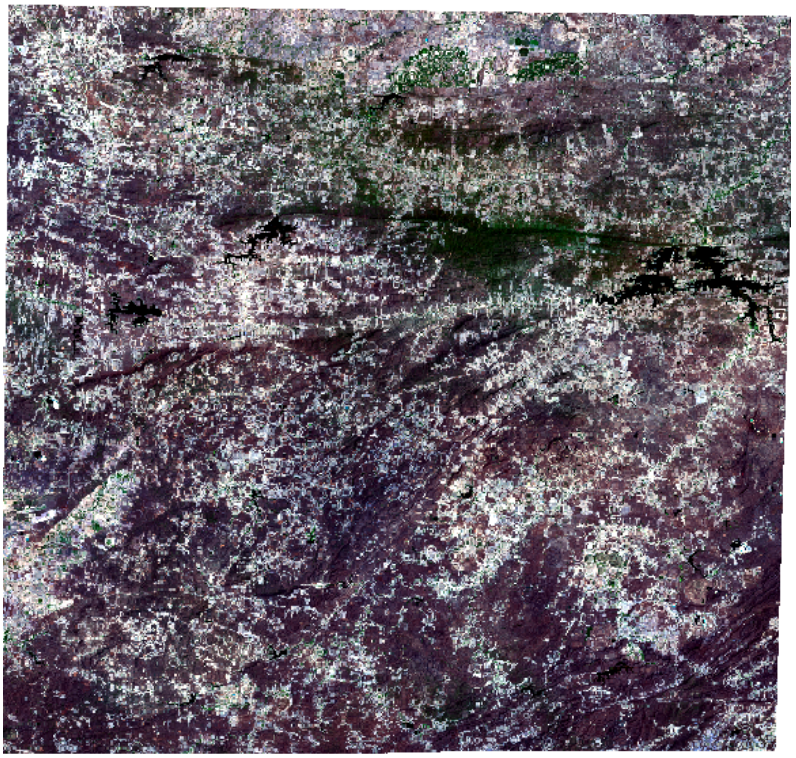

Figura 02: Imagem Landsat 8, com a composição das bancas B2, B3, B4 e B5

Inicialmente foi analisada o coeficiente de correlação entre as quatro bandas espectrais selecionadas da imagem Landsat, cujo resultado obtido apresenta-se na tabela abaixo:

	B2	B3	B4	B5
B2	1.0000000	0.9652100	0.9705477	0.6750086
B3	0.9652100	1.0000000	0.9750729	0.7555930
B4	0.9705477	0.9750729	1.0000000	0.7007850
B5	0.6750086	0.7555930	0.7007850	1.0000000

Tabela 01: Correlação em as bancas B2, B3, B4 e B5

A diagonal principal apresenta a correlação entre as mesmas bandas, e consequentemente seu valor é igual a 1. A parte superior da tabela é espelhada em relação à inferior, pois resulta da mesma combinação de bandas. Com os valores encontrados pode-se analisar melhor a correlação entre as bandas espectrais e seu grau de relacionamento. Nota-se que as bandas do visível (B2, B3 e B4) apresentam alto grau de correlação positiva entre si, com todos os valores superiores a 0,9. Havendo assim um grau de redundância entre os dados, devido à alta correlação entre eles. Esse comportamento pode estar associado ao fato dessas bandas estarem posicionadas em faixas próximas do espectro eletromagnético e pela presença de alvos com assinaturas espectrais similares na área de estudo. O que não ocorre com os coeficientes de correlação entre a banda do infravermelho e as bandas do visível, que apresenta valores mais baixos, o que aponta para uma

correlação mais fraca. Tal comportamento é confirmado na análise do espectro apresentado abaixo.

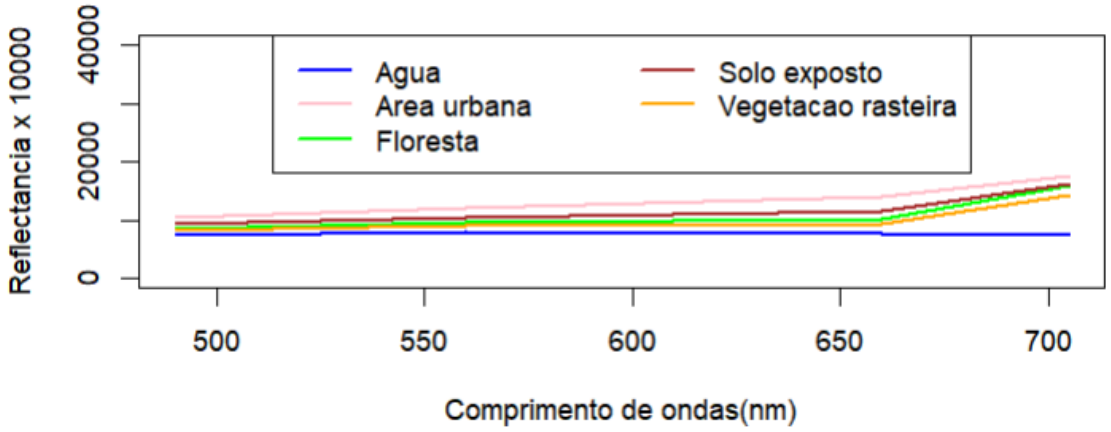

Figura 03: Assinatura espectral

Assim, a partir dos resultados acima evidencia-se o alto grau de correlação entre as bandas espectrais, principalmente da faixa do visível, que produz redundância entre as informações que pode dificultar o processo de classificação de imagens.

Nesse momento, a técnica PCA foi aplicada sobre as bandas selecionadas, sendo produzidas quatro componentes principais, cujos resultados foram descritos através da Tabela 2, que apresenta a porcentagem de variância original explicada para cada componente principal e os seus autovetores correspondentes.

	CP1	CP2	CP3	CP4
B2	-0.5143036	-0.33166501	0.79049220	0.02494037
B3	-0.5250155	-0.1438086	-0.3782386	-0.74871328
B4	-0.5189867	-0.2750859	-0.47377245	0.65614038
B5	-0.4364668	0.08908732	0.08692239	0.09102864
Variância Explicada	88.44%	10.26%	0.77%	0.54%

Tabela 02: Variância original explicada para cada componente principal e autovetores

Desta forma, a primeira componente principal detém a maior percentagem de variância explicada (87,25%), seguida pela segunda que possui 12,30%, somando 99,55% da variância total do banco de dados. Por outro lado, a terceira componente explica 0,28% e a última 0,17%, sendo que a última está provavelmente relacionada com os ruídos existentes na imagem original. Assim, quando observados os autovetores apresentados na Tabela 2 nota-se que a 1º CP apresenta maior contribuição positiva da banda do infravermelho próximo (B5), seguida da banda B3 com contribuição também positiva e as demais bandas contribuem negativamente.

No gráfico abaixo pode-se visualizar a distribuição dos pontos em função das componentes principais 1 e 2 e a representação gráfica biplot dos dois primeiros componentes principais.

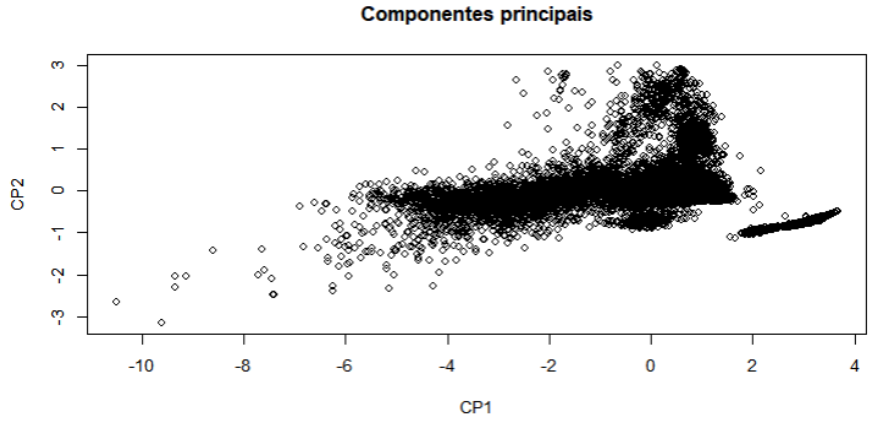

Figura 05: Distribuição dos dados

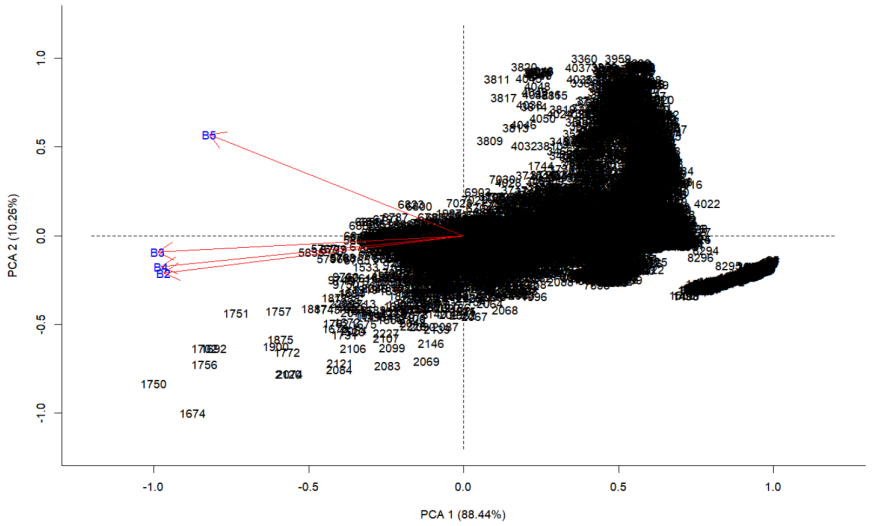

Figura 04: Gráfico biplot das componentes CP1 e CP2

Na etapa seguinte foi realizada a classificação de imagens utilizando as bandas espectrais originais, de maneira supervisionada e com uma abordagem pixel-a-pixel a partir de dois classificadores distintos: Random Forest e SVM (Support Vector Machine), a partir de amostra estabelecida por 100 pontos de dados aleatórios, partir da análise visual da imagem (Figura 2), com as informações das bancas B2, B3, B4 e B5, obtidos com o uso do software Qgis. A fim de comparar se há algum incremento na qualidade da classificação ao usar as componentes principais em relação às bandas espectrais. Sendo ao total escolhidas cinco classes distintas: Água (20 dados), Área Urbana (15 dados), Floresta (34 dados), Solo Exposto (23 dados) e Vegetação Rasteira (8 dados).

A amostra de 100 dados utilizada na análise, foram particionadas em treinamento do algoritmo de classificação (70%) e no teste da acurácia do classificador (30%).

Em seguida foi realizado a classificação da imagem, para isso utilizou-se dois classificadores distintos, o Random Forest e SVM, com os dados originais da imagem Landsat e os dados transformados pelo método PCA, visando a comparação a potencialidade da técnica de Análise de Componentes Principais para o aprimoramento da acurácia da classificação do uso e cobertura do solo.

O classificador Random Forest utiliza um conjunto de árvores de decisão para realizar uma previsão. As árvores de decisão são criadas independentemente, através de um subconjunto de amostras de treinamento.

O classificador SVM é uma técnica de classificação supervisionada de imagens, baseada em um algoritmo de otimização que define, através das amostras de treinamento, planos de separação ótimos entre as classes a fim de maximizar a distância entre elas.

Após a aplicação do algoritmo de o Random Forest, foi realizada a análise da influência no modelo proveniente das bandas espectrais através do gráfico de diminuição de média de precisão e da diminuição média no coeficiente de Gini.

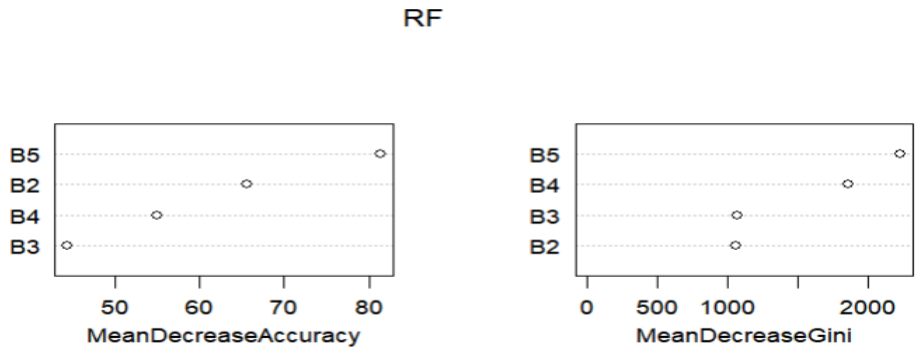

Figura 06:Gráfico de diminuição de média de precisão à esquerda e da diminuição média no coeficiente de Gini a direita.

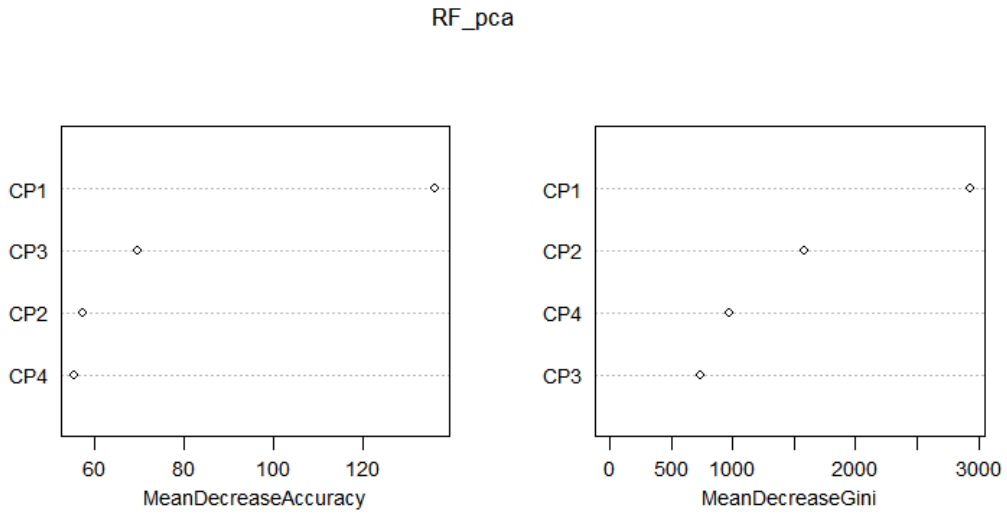

Figura 07:Gráfico de diminuição de média de precisão à esquerda e da diminuição média no coeficiente de Gini a direita aplicado a PCA.

O gráfico Mean Decrease Accuracy expressa quanta precisão o modelo perde ao excluir cada variável. Quanto mais a precisão é prejudicada, mais importante é a variável para o sucesso da classificação. As variáveis são apresentadas em ordem decrescente de importância. A diminuição média no coeficiente de Gini é uma medida de como cada variável contribui para a homogeneidade dos nós e folhas na floresta aleatória resultante. Quanto maior o valor da diminuição média da precisão ou da diminuição média do escore de Gini, maior será a importância da variável no modelo. Desta forma, pode-se constatar a influência da banda B5 na modelagem sem a aplicação do PCA.

Pensando que toda a classificação de uso e cobertura da terra possui erros, a quantificação e comunicação destes erros ao público é uma etapa muito importante do trabalho, sendo assim nas Tabelas a seguir será representa a matriz de confusão da classificação com o algoritmo Random Forest e a SVM dados originais da imagem Landsat e os dados transformados pelo método PCA para a identificação do melhor modelo de predição.

```
> print(CM.RF)
Confusion Matrix and Statistics

                  Reference
Prediction          Agua Area urbana Floresta Solo exposto Vegetacao rasteira
  Agua               455          0        0           0                  1
  Area urbana          0        173        1          29                  0
  Floresta             0          0      578         124                 37
  Solo exposto         0         81      104        1295                 37
  Vegetacao rasteira   0          0       60          30                624

Overall Statistics

               Accuracy : 0.8611
                 95% CI : (0.8494, 0.8722)
    No Information Rate : 0.4073
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8099

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Agua Class: Area urbana Class: Floresta Class: Solo exposto Class: Vegetacao rasteira
Sensitivity               1.0000           0.68110          0.7779              0.8762                    0.8927
Specificity               0.9997           0.99111          0.9442              0.8968                    0.9693
Pos Pred Value            0.9978           0.85222          0.7821              0.8537                    0.8739
Neg Pred Value            1.0000           0.97636          0.9429              0.9134                    0.9743
Prevalence                0.1254           0.06999          0.2047              0.4073                    0.1926
Detection Rate            0.1254           0.04767          0.1593              0.3568                    0.1719
Detection Prevalence      0.1257           0.05594          0.2036              0.4180                    0.1967
Balanced Accuracy         0.9998           0.83611          0.8611              0.8865                    0.9310
```

Figura 08: Matriz de confusão do algoritmo Random Forest

A tabela acima, apresenta a matriz de confusão da classificação com o algoritmo Random Forest com os dados originais da imagem. Pode-se observar que a acurácia geral desta classificação foi de 86,11%, com índice Kappa de 8,099 e erro de estimativa aproximadamente de 14,13%.

```
> print(CM.SVM)
Confusion Matrix and Statistics

                  Reference
Prediction          Agua Area urbana Floresta Solo exposto Vegetacao rasteira
  Agua               455          0        1           0                  0
  Area urbana          0        111        0          17                  0
  Floresta             0          0      134           0                 10
  Solo exposto         0        143      529        1440                160
  Vegetacao rasteira   0          0       79          21                529

Overall Statistics

               Accuracy : 0.7355
                 95% CI : (0.7208, 0.7498)
    No Information Rate : 0.4073
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.614

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Agua Class: Area urbana Class: Floresta Class: Solo exposto Class: Vegetacao rasteira
Sensitivity               1.0000           0.43701          0.18035              0.9743                    0.7568
Specificity               0.9997           0.99496          0.99653              0.6132                    0.9659
Pos Pred Value            0.9978           0.86719          0.93056              0.6338                    0.8410
Neg Pred Value            1.0000           0.95915          0.82525              0.9720                    0.9433
Prevalence                0.1254           0.06999          0.20474              0.4073                    0.1926
Detection Rate            0.1254           0.03059          0.03692              0.3968                    0.1458
Detection Prevalence      0.1257           0.03527          0.03968              0.6261                    0.1733
Balanced Accuracy         0.9998           0.71599          0.58844              0.7937                    0.8613
```

Figura 09: Matriz de confusão do algoritmo SVM

Já para a classificação utilizando o algoritmo SVM com dados originais, observa-se que a acurácia geral foi de 73,55% e índice Kappa de 0,614.

```
> print(CM.RF_pca)
Confusion Matrix and Statistics

                    Reference
Prediction           Agua Area urbana Floresta Solo exposto Vegetacao rasteira
  Agua                454           0        2           0                  0
  Area urbana           0         186        0          29                  0
  Floresta              0           0      585         105                 30
  Solo exposto          1          68      106        1316                 41
  Vegetacao rasteira    0           0       50          28                628

Overall Statistics

               Accuracy : 0.8732
                 95% CI : (0.862, 0.8839)
    No Information Rate : 0.4073
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8264

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Agua Class: Area urbana Class: Floresta Class: Solo exposto Class: Vegetacao rasteira
Sensitivity               0.9978            0.73228          0.7873              0.8904                    0.8984
Specificity               0.9994            0.99141          0.9532              0.8996                    0.9734
Pos Pred Value            0.9956            0.86512          0.8125              0.8590                    0.8895
Neg Pred Value            0.9997            0.98008          0.9457              0.9227                    0.9757
Prevalence                0.1254            0.06999          0.2047              0.4073                    0.1926
Detection Rate            0.1251            0.05125          0.1612              0.3626                    0.1731
Detection Prevalence      0.1257            0.05924          0.1984              0.4222                    0.1945
Balanced Accuracy         0.9986            0.86185          0.8703              0.8950                    0.9359
>
```

Figura 10: Matriz de confusão do algoritmo Random Forest utilizando PCA

Para a classificação utilizando o algoritmo Random Forest com dados transformados, observa-se que a acurácia geral foi de 87,5%, índice Kappa de 0,8289 e erro de estimativa de aproximadamente de 12,32%. Nota-se desta forma um aumento na acurácia geral se comparado as duas classificações anteriormente apresentadas.

```
> print(CM.SVM_pca)
Confusion Matrix and Statistics

                  Reference
Prediction          Agua Area urbana Floresta Solo exposto Vegetacao rasteira
  Agua               455           0        0           0                  0
  Area urbana          0         159        1          12                  0
  Floresta             0           0      164           4                 11
  Solo exposto         0          95      515        1439                160
  Vegetacao rasteira   0           0       63          23                528

Overall Statistics

               Accuracy : 0.7564
                 95% CI : (0.7421, 0.7703)
    No Information Rate : 0.4073
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6472

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Agua Class: Area urbana Class: Floresta Class: Solo exposto Class: Vegetacao rasteira
Sensitivity               1.0000            0.62598         0.22073              0.9736                     0.7554
Specificity               1.0000            0.99615         0.99480              0.6420                     0.9706
Pos Pred Value            1.0000            0.92442         0.91620              0.6514                     0.8599
Neg Pred Value            1.0000            0.97252         0.83217              0.9725                     0.9433
Prevalence                0.1254            0.06999         0.20474              0.4073                     0.1926
Detection Rate            0.1254            0.04381         0.04519              0.3965                     0.1455
Detection Prevalence      0.1254            0.04740         0.04932              0.6087                     0.1692
Balanced Accuracy         1.0000            0.81107         0.60776              0.8078                     0.8630
>
```

Figura 11: Matriz de confusão do algoritmo SVM utilizando PCA

Na classificação utilizando o algoritmo SVM com dados transformados, observa-se que a acurácia geral foi de 75,6% e índice Kappa de 0,6466. Também constata-se um aumento na acurácia desta classificação em comparação com a realizada com o mesmo algoritmo porém sem a utilização no PCA.

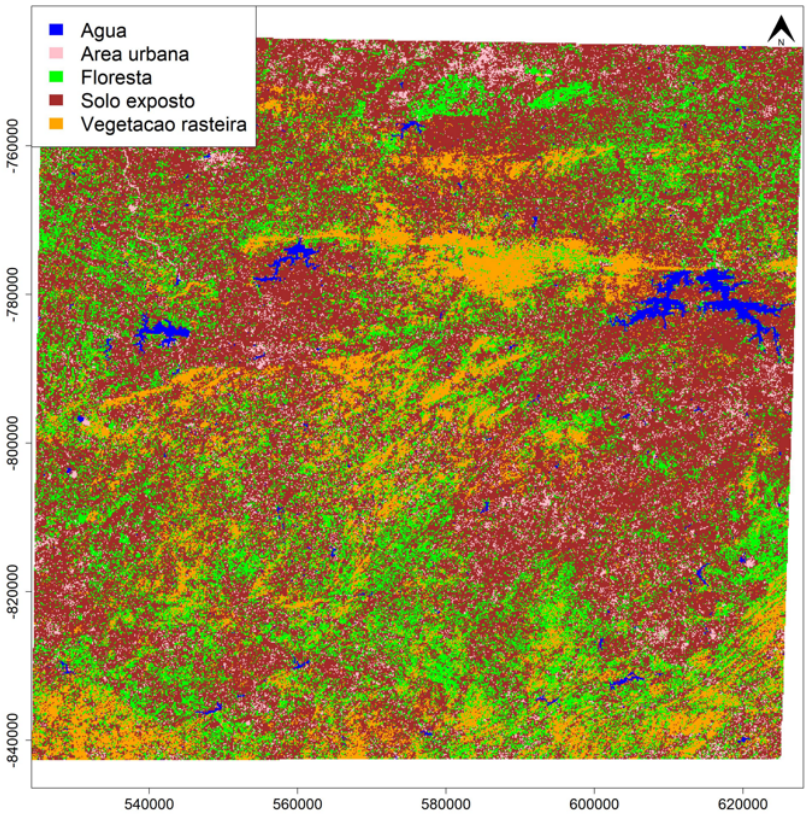

Figura 12: Imagem gerada com o algoritmo Random Forest

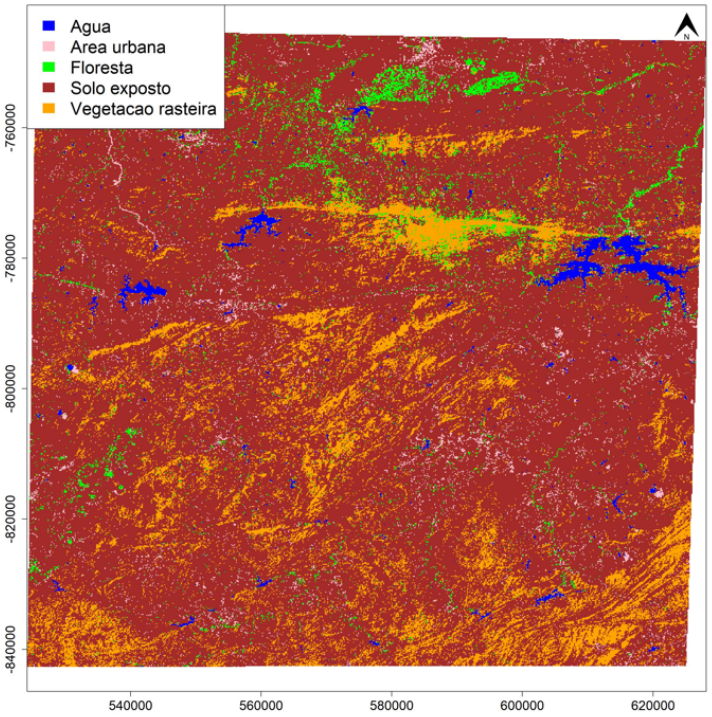

Figura 12: Imagem gerada com o algoritmo SVM

CONCLUSÃO

Neste estudo de caso pode-se constatar que a utilização de componentes principais para a classificação de imagem visando o mapeamento do uso e cobertura da terra se mostrou eficaz, aumentando a acurácia e reduzindo os erros. Pode-se concluir, então, que o uso de componentes principais apresenta potencialidade para gerar melhores resultados na classificação de uso e cobertura da terra quando comparados ao uso das bandas espectrais.

Neste sentido, a Análise de Componentes Principais aplicada a imagens de sensoriamento remoto, é bastante significante uma vez que o método parte do princípio de que as bandas espectrais utilizadas estão correlacionadas entre si, havendo redundância de dados. Assim, a PCA condensa a informação disponível em número menor de canais espectrais, totalmente descorrelacionados, realçando os alvos na imagem e melhorando o resultado da classificação. Isso ocorre, pois, a utilização de dados não redundantes é de muito interesse no processo de classificação de imagens.

Sendo assim, a aplicação da PCA no campo do sensoriamento remoto é vantajosa devido a sua capacidade de identificar padrões nos dados, destacando as diferenças e similaridades. A técnica resulta em baixa perda de informação, com concentração das informações mais relevantes nas primeiras componentes, permitindo diminuir a demanda e tempo computacional pela redução do número de bandas e pela remoção da correlação.

#################### **ANEXO** ####################

```
# Carregar imagens
img = stack("dados/Recorte/B2_recorte.tif",
               "dados/Recorte/B3_recorte.tif",
               "dados/Recorte/B4_recorte.tif",
               "dados/Recorte/B5_recorte.tif")
names(img) = c("B2","B3", "B4", "B5")
print(img)
plotRGB(img, r = 4, g = 3, b = 2,axes = T, stretch = 'lin',
main = "Landsat 2 cor verdadeira")
writeRaster(x=img, filename=
"dados/Recorte/Imagem_original_all.tif")
### Carregando dados amostrais da cobertura do solo
amostra = readOGR("dados/Recorte/Amostra/Dados amostrais.shp")
View(data.frame(amostra))
### Juntar as feições de cada classe da amostra
unidos_shp = gUnaryUnion(spgeom = amostra, id =
amostra$Classes, checkValidity = 2L)
unidos_shp
## extrair as amostras unidas e criar um date.frame para a
utilização do treinamento posteriormente
atributo = extract(x = img, y = unidos_shp)
## criar um data.frame para cada classe utilizando o atributo
criado
names(unidos_shp)
agua = data.frame(Classe = "Agua", atributo[1])
area_urbana = data.frame(Classe = "Area urbana", atributo[2])
floresta = data.frame(Classe = "Floresta", atributo[3])
solo = data.frame(Classe = "Solo exposto", atributo[4])
veg_rasteira = data.frame(Classe = "Vegetacao rasteira",
atributo[5])
### juntar todas a data.frame em uma unica
amostras_final = rbind(agua, area_urbana,floresta,solo,
veg_rasteira)
write.csv(amostras_final,
"dados/Recorte/Amostra/amostra_final.csv")
# Contar a quantidade de dados por categoria
contagem = table(amostra$Classes)
# Mostrar a contagem
print(contagem)
## calcular o expectro de reflectancia de cada classe
```

```
##agrupar classe
agrupado = group_by(amostras_final, Classe)
print(agrupado)
### media de cada classes
media_ref = summarise_each(agrupado, mean)
print(media_ref)
### calculo do especto
refs = t(media_ref [ ,2:5])
cores = c ("blue","pink", "green","brown", "orange")
comp_onda = c(490, 560, 660, 705)
matplot(x= comp_onda, y=refs, type = "l", lwd = 2, lty = 1,
xlab = "Comprimento de ondas(nm)", ylab = "Reflectancia x
10000", col = cores, ylim = c(0,40000))
legend('top', legend = media_ref$Classe, col = cores, lty = 1,
ncol = 2, lwd = 2)
### calculo da matrix de correlação entre as bandas
# Extrair os valores das bandas e criar uma matriz de dados
data_matrix = data.frame (amostras_final$B2,
amostras_final$B3, amostras_final$B4, amostras_final$B5)
names(data_matrix) = c("B2","B3", "B4", "B5")
# Calcular a matriz de correlação
cor_matrix = cor(data_matrix)

# Imprimir a matriz de correlação
print(cor_matrix)
# Carregar os arquivos de treino e validação
# Checar a class (strucutre dos meus dados não pode ser texto)
str(amostras_final)
amostras_final$Classe = as.factor(amostras_final$Classe)
#Separação dos dados em treinamento e validação
set.seed(1234) # mantem o mesmo resultado
amostras_treino = sample.split(amostras_final$Classe,
SplitRatio = 0.7)
(amostras_treino)
##Separar em dados de treino e teste (criar o dataframe de
treino e teste)
train = amostras_final[amostras_treino,]
valid = amostras_final[amostras_treino == F, ]
write.csv(train, "dados/Recorte/Amostra/Amostras_treino.csv")
write.csv(valid, "dados/Recorte/Amostra/Amostras_teste.csv")
### Classificação pelo RandomForest
train$Classe = as.factor(train$Classe)
```

```r
valid$Classe = as.factor(valid$Classe)
set.seed(1234) # para gerar arvore de decisão aleatoria com
mesmo resultado
RF = randomForest(Classe~., data = train, ntree = 100, mtry =
3, importance = T) # classe relacionada com todas as bandas
varImpPlot(RF) # O gráfico Mean Decrease Accuracy expressa
quanta precisão o modelo perde ao excluir cada variável.
Quanto mais a precisão é prejudicada, mais importante é a
variável para o sucesso da classificação. As variáveis são
apresentadas em ordem decrescente de importância. A diminuição
média no coeficiente de Gini é uma medida de como cada
variável contribui para a homogeneidade dos nós e folhas na
floresta aleatória resultante. Quanto maior o valor da
diminuição média da precisão ou da diminuição média do escore
de Gini, maior será a importância da variável no modelo.
importance(RF)
# Support Vector Machines - SVM
set.seed(1234)
SVM = svm(Classe~., kernel = 'polynomial', data = train)
## Validação dos modelos
pred.RF = predict(RF, valid)
pred.SVM = predict(SVM, valid)
## Criacao da matriz de confusão
CM.RF = confusionMatrix(data = pred.RF, reference =
valid$Classe)
CM.SVM = confusionMatrix(data = pred.SVM, reference =
valid$Classe)
print(CM.RF)
print(CM.SVM)
#Salvando os modelos
saveRDS(object = RF, file =
"dados/Recorte/Classsificacao_rf.rds")
saveRDS(object = SVM, file =
"dados/Recorte/Classsificacao_SVM.rds")
############################################################
# Usando PCA
# Calcular a Matriz de covariância
Bandas = data.frame (amostras_final$B2, amostras_final$B3,
amostras_final$B4, amostras_final$B5)
nomes_colunas = c("B2", "B3", "B4", "B5")
colnames(Bandas) =  nomes_colunas # alterar o nome das colunas
print("Bandas:")
```

```r
print(Bandas)
Bandas_centralizados = scale(Bandas) # Padronização # a média
de cada variável foi subtraída transformando para toda escala
cov_matrix = cov(Bandas_centralizados) #calcula a matriz de
covariância de um conjunto de dados. A matriz de covariância é
uma medida estatística que descreve como as variáveis em um
conjunto de dados mudam juntas
print("Matriz de Covariância:")
print(cov_matrix)
##PCA = PCA(Bandas_centralizados)
# Calcular autovalores e autovetores
eigen_resultados = eigen(cov_matrix) # Os autovetores são os
vetores próprios que definem as direções dos componentes
principais.
autovalores = eigen_resultados$values # Os autovalores
representam a quantidade de variância explicada por cada
componente principal
autovetores = eigen_resultados$vectors # Os autovetores são os
vetores próprios que definem as direções dos componentes
principais.
# Ordenar autovalores e autovetores
ordem = order(autovalores, decreasing = TRUE)
autovalores = autovalores[ordem]
autovetores = autovetores[, ordem]
print("Autovalores:")
print(autovalores)
print("Autovetores:")
print(autovetores)
# Calcular a porcentagem da variância retida para cada vetor
percentage_retained_CP1 = (autovalores[1] / sum(autovalores))
* 100
percentage_retained_CP2 = (autovalores[2] / sum(autovalores))
* 100
percentage_retained_CP3 = (autovalores[3] / sum(autovalores))
* 100
percentage_retained_CP4 = (autovalores[4] / sum(autovalores))
* 100
# Tabela Autovetores e Porcentagem de Variância Explicada para
as Componentes Principais
Autovetores = c("B2", "B3", "B4", "B5")
CP1 = c(autovetores[, 1])
CP2 = c(autovetores[,2 ])
```

```
CP3 = c(autovetores[,3 ])
CP4 = c(autovetores[, 4])
Variancia_Explicada = c(percentage_retained_CP1,
percentage_retained_CP2,
                        percentage_retained_CP3,
percentage_retained_CP4)
Tabela_resumo = data.frame(Autovetores, CP1, CP2, CP3, CP4,
Variancia_Explicada )
print(Tabela_resumo)
# Escolha dos componentes principais que serão utilizados
componentes_principais = autovetores[,1:4] # usando todas as
componentes
# Transformar os dados
dados_transformados = Bandas_centralizados %*%
componentes_principais
# Gráfico dos pontos com relação a CP1 e CP2
plot (dados_transformados[,1], dados_transformados[,2],
xlab="CP1",  ylab="CP2", main = "Componentes principais") #
plotagem de dados das 2 primeiras componentes
# Obtendo a correlação dos dados padronizados com os CPs
cor_dp_cp = cor(Bandas_centralizados, dados_transformados[,
1:2])
nomes_col_cp = c("CP1", "CP2")
colnames(cor_dp_cp) =  nomes_col_cp
print(cor_dp_cp)
# Criar o gráfico biplot
plot(1, type = "n", xlab = "", ylab = "", xlim = c(-1, 1),
ylim = c(-1, 1))
for (i in seq_along(autovalores[1:4])) {
  arrows(0, 0, autovetores[i,1], autovetores[i,2], angle = 20,
length = 0.1, col = "red")
} # Adicionar vetores para as variáveis originais
text(autovetores[i, 1], autovetores[i, 2], labels =
colnames(dados_transformados ), pos = 3, cex = 0.7, col =
"blue") # Adicionar rótulos às observações
#Separação dos dados em treinamento e validação
dados_pca = data.frame(amostras_final$Classe,
dados_transformados)
nomes_colunas = c("Classe", "CP1", "CP2", "CP3", "CP4")
colnames(dados_pca) =  nomes_colunas # alterar o nome das
colunas
dados_pca = dados_pca
```

```
str(dados_pca) #verificando se todos os dados são numeros
set.seed(1234) # mantem o mesmo resultado
dados_pca_treino = sample.split(dados_pca$Classe, SplitRatio =
0.7)
##Separar em dados de treino e teste (criar o dataframe de
treino e teste)
train_pca = dados_pca[dados_pca_treino,]
valid_pca = dados_pca[dados_pca_treino == F, ]
write.csv(train,
"dados/Recorte/Amostra/dados_pca_treino1.csv")
write.csv(valid, "dados/Recorte/Amostra/dados_pca_teste1.csv")
### Classificação pelo RandomForest
train_pca$Classe = as.factor(train_pca$Classe)
valid_pca$Classe = as.factor(valid_pca$Classe)
set.seed(1234) # para gerar arvore de decisão aleatoria com
mesmo resultado
RF_pca = randomForest(Classe~., data = train_pca, ntree = 100,
mtry = 3, importance = T) # classe relacionada com todas as
bandas
varImpPlot(RF_pca) # O gráfico Mean Decrease Accuracy  e
diminuição média no coeficiente de Gini
importance(RF_pca)
# Support Vector Machines - SVM
set.seed(1234)
SVM_pca = svm(Classe~., kernel = 'polynomial', data =
train_pca)
## Validação dos modelos
pred.RF_pca = predict(RF_pca, valid_pca)
pred.SVM_pca = predict(SVM_pca, valid_pca)
## Criacao da matriz de confusão
CM.RF_pca = confusionMatrix(data = pred.RF_pca, reference =
valid_pca$Classe)
CM.SVM_pca = confusionMatrix(data = pred.SVM_pca, reference =
valid_pca$Classe)
print(CM.RF_pca)
print(CM.SVM_pca)
print(CM.RF)
print(CM.SVM)
#############################################################
## predição para o raster
RF.raster = predict(img,RF)
SVM.raster = predict(img,SVM)
```

```
RF_pca.raster = predict(img,RF_pca)
SVM_pca.raster = predict(img,SVM_pca)
# Plotagem colorida
#plotagem da imagem RF
cores = c ("blue","pink", "green","brown", "orange")
classes = c ("Agua", "Area urbana", "Floresta", "Solo
exposto", "Vegetacao rasteira")
jpeg(filename = "dados/Recorte/classificacao1.jpeg", width =
15, height = 15, res = 200, units = 'in')
plot(RF.raster, legend = FALSE, col = cores, main =
"Classificação RF",
     cex.axis = 1.5, cex.main = 1.5)
legend('topleft', legend = classes, fill = cores, border =
FALSE, cex =2)
addnortharrow(cols = c("black", 'black'), scale = 0.755)
dev.off()
#plotagem da imagem SVM
cores = c ("blue","pink", "green","brown", "orange")
classes = c ("Agua", "Area urbana", "Floresta", "Solo
exposto", "Vegetacao rasteira")
jpeg(filename = "dados/Recorte/classificacao_SVM.jpeg", width
= 15, height = 15, res = 200, units = 'in')
plot(SVM.raster, legend = FALSE, col = cores, main =
"Classificação SVM",
     cex.axis = 1.5, cex.main = 1.5)
legend('topleft', legend = classes, fill = cores, border =
FALSE, cex =2)
addnortharrow(cols = c("black", 'black'), scale = 0.755)
dev.off()
#plotagem da imagem RF_pca
cores = c ("blue","pink", "green","brown", "orange")
classes = c ("Agua", "Area urbana", "Floresta", "Solo
exposto", "Vegetacao rasteira")
jpeg(filename = "dados/Recorte/classificacao_RFpca1.jpeg",
width = 15, height = 15, res = 200, units = 'in')
plot(RF_pca.raster, legend = FALSE, col = cores, main =
"Classificação RF_pca",
     cex.axis = 1.5, cex.main = 1.5)
legend('topleft', legend = classes, fill = cores, border =
FALSE, cex =2)
addnortharrow(cols = c("black", 'black'), scale = 0.755)
dev.off()
```

```
#plotagem da imagem SVM_pca
cores = c ("blue","pink", "green","brown", "orange")
classes = c ("Agua", "Area urbana", "Floresta", "Solo
exposto", "Vegetacao rasteira")
jpeg(filename = "dados/Recorte/classificacao_SVM_pca.jpeg",
width = 15, height = 15, res = 200, units = 'in')
plot(SVM_pca.raster, legend = FALSE, col = cores, main =
"Classificação SVM_pca",
     cex.axis = 1.5, cex.main = 1.5)
legend('topleft', legend = classes, fill = cores, border =
FALSE, cex =2)
addnortharrow(cols = c("black", 'black'), scale = 0.755)
dev.off()
```